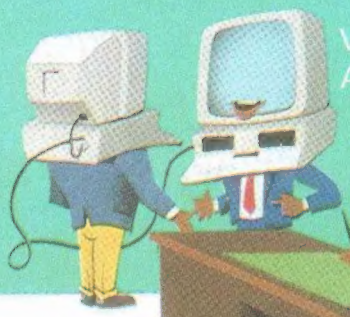
For Tandy's 100 and 200 Portables and 1000, 1200 and 2000 MS-DOS Computers

PCM

The Personal Computing Magazine for Tandy® Computer Users

Vol. III No. 2
August 1985
U.S. $3
CDN $3.95



ViaNet

Tandy's New Omnicentric Communications System for Your Office

Access MS-DOS Commands From BASIC

Create Forms with the Aid of a Spreadsheet

Select Colors for Your 2000 — *John Harrell Shows you how*

plus:
Bill Barden on Sequential Files
Randy Graham on Graphics Transmission
and more product reviews and bar code listings

FRED W. CRAWFORD
© 1985

8607 413
<<PCM>>
KAREN K OKADA
4380 OGBURN AVE
WINSTON-SALEM NC 27105

SOFTLE

TELECOMMUTER™

INTEGRATED WORD PROCESSING AND COMMUNICATIONS SOFTWARE

For the Tandy 1000, 1200 and 2000, and IBM-PC/XT/AT/jr and compatible computers. Uses the same editing and communications commands as the Model 100/200 — plus much, much more!

EASY TO USE . . .

"I am writing to let you know that I like your program very much indeed. The high speed transfers from my Model 100 to my Tandy 1000 are flawless. So many word processors are encumbered by in my view unnecessary baggage for the normal user. Yours is powerful, yet the user interface is elegant in its simplicity. It is essentially a 'Super Model 100'. Your advertising does not express the versatility of the program. I was intending to use it for general writing and articles. I now intend to use it for a book I'm working on."

Jay S. Moynihan, Esq.
Ashland, WI

WITH POWER FOR THE BIG JOBS!

"A few weeks ago I received Telecommuter . . . it has been in constant use as a word processor, a telecommunications package (which has allowed me to access numerous databases), and to transfer files from my Model 100 to the Tandy 2000. Since Telecommuter has arrived, both Word Perfect and Videotex Plus . . . the word processor and terminal program I had been using . . . have not left the shelf. I have also used Telecommuter to write dBase II command files, and it has made that job so much easier than when utilizing the programming section of dBase itself. My most enthusiastic congratulations on a superior product."

Ivan K. Goldberg, M.D.
New York, NY

Available in six versions:

TC-Write It™ $125, **TC-Standard** and **TC-XMODEM** $200, **TC-DeLuxe** $300, **TC-plus** $400, **TC-Corporate** $1000.

Advanced features include DOS access from Word Processor, XMODEM file transfer, Telcom macros, and full VT100 terminal emulation. Host Mode gives control of your computer to a remote Model 100/200 or MS-DOS computer running Telecommuter.

Call or write today for more information on the entire Telecommuter product line. Product upgrades available for difference in price. We accept MasterCard and Visa for mail or telephone orders, and we ship free in the USA.

Look for Telecommuter at your local Radio Shack Computer Store!

SIGEA SYSTEMS, INC.

19 Pelham Road, Weston, MA 02193 (617) 647-1099

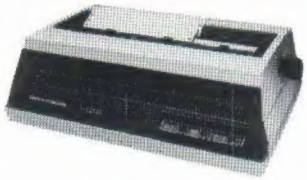
From Computer Plus to YOU . . .

PLUS after PLUS after PLUS

Tandy 200 24K $739
Model 100 24K $510

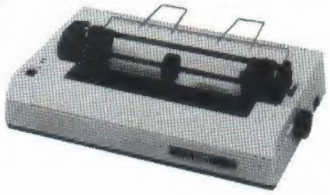
Tandy 1200 HD $1599
10 Meg. 256K

Tandy 1000 $779
Tandy 1000 HD $1539



Smith Corona L-1000
$259

Color Computer II
w/16K Ext. Basic $179



DMP105 $160

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Tandy 1000 1 Drive 128K	779.00
Tandy 1000 HD 10 Meg. 256K	1539.00
Tandy 1200 10 Meg. 256K	1599.00
Tandy 2000 2 Drive 256K	1295.00
Tandy 2000 10 Meg. 256K	2025.00
Model IV Desktop 64K	959.00
PRINTERS	
Radio Shack DMP-2100P	1159.00
Radio Shack DMP-430	660.00
Radio Shack DWP-220 Daisy Wheel	469.00
Radio Shack TRP-100 Portable	229.00
Silver Reed EXP-550 Daisy Wheel	430.00
Star SG-10	245.00
Star SG-15	410.00
Star SD-10	365.00
Panasonic P-1091	275.00
Toshiba 351	1325.00
CITOH Prowriter 8510AP + NLQ	345.00
Okidata 192	379.00
Okidata 193	539.00
Epson LX-80	245.00
Epson FX-80	359.00

Tandy 1000/1200 ACCESSORIES	
Tandy 1000 Disk Drive Kit	159.00
Tandy 1000 10 Meg. Hard Drive	579.00
Hard Drive Controller Board	265.00
256K Ram Board (inc. 128K & DMA)	199.00
512K Ram Board (includes 128K)	169.00
256K Memory Plus Expansion Brd.	249.00
PBJ Multi-Function Board (128K)	269.00
PBJ Multi-Function Board (256K)	299.00
PBJ Multi-Function Board (512K)	349.00
128K Ram Upgrade Kit (Tandy)	115.00
128K Ram Upgrade Kit (NEC)	78.00
256K Ram Upgrade (for PBJ Board)	69.00
300 Baud Modem Board	129.00
RS-232 Serial Board	89.00
Digi-Mouse/Clock Board	89.00
Tandy 1000/1200 MONITORS	
Tandy VM-2 Green Monitor	129.00
Tandy CM-2 RGB Color Monitor	379.00
Tandy VM-3 TTL Green Monitor	179.00
Taxan 122 TTL Amber Monitor	139.00
Tandy 1200 Text Monitor Adapt.	179.00
Tandy 1200 Color/Graphics Adapt	240.00

MODEMS	
Radio Shack DCM-3 Model	52.00
Radio Shack DC Modem II	160.00
Radio Shack DC Modem 2212	315.00
Hayes Smartmodem II 300 Baud	215.00
Hayes Smartmodem 1200 Baud	429.00
Novation J-Cat 300 Baud	115.00
FOR EVERYONE	
Network Four Outlet Surge Prot.	69.95
CCR-81 Cassette Recorder	52.00
CCR-82 Cassette Recorder	43.00
C-20 Digital Cassette Tapes (10pk)	9.95
Verbatim SSDD Plastic Box (10pk)	24.95
Verbatim DDDD Plastic Box (10pk)	29.95
Generic SSDD Diskettes (10pk)	17.95
NEC 64K Ram Chips (set of 8)	49.00
Radio Shack 16K Ram Chips (8)	25.00
Locking Disk Box (holds 70)	19.95
Flip N File Disk 50	15.00
Fanfold CleanEdge Paper (2600)	35.00

Radio Shack software 10% off.
Send for complete listing of
brand name software and hardware.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA AMERICAN EXPRESS MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

TRS-80 is a registered trademark of Tandy Corp.

PCM

The Personal Computing Magazine for Tandy® Computer Users

Vol. III, Issue 2 August 1985

Editor and Publisher
Lawrence C. Falk

Managing Editor Courtney Noe
Senior Editor Jim Reed
Technical Editor Danny Humphress
Submissions Editor Jutta Kapfhammer
Copy Editor Tamara Dunn
Reviews Editor Monica Dorth
Editorial Assistants Jody Doyle, Wendy Falk, Debbie Hartley, Judi Hutchinson, Angela Kapfhammer, Belinda Kirby, Suzanne Benish Kurowsky, Shirley Morgan, Kevin Nickols
Editorial Consultant Vince Lord
Contributing Editors William Barden, Jr., Randy Graham, Jim Hawk, Richard White
Art Director Jerry McKiernan
Assistant Art Director Sally Gellhaus
Designers Tracey Jones, Heidi Maxedon, Kevin Quiggins
Technical Assistant Ed Ellers
Advertising Coordinator Doris Taylor (502) 228-4492
Advertising Assistant Debbie Baxter
General Manager Patricia H. Hirsch
Assistant General Manager for Finance Donna Shuck
Bookkeeper Diane Moore
Advertising Accounts Beverly Taylor
Dealer Accounts Judy Quashnock
Administrative Assistant To The Publisher Marianne Booth
Manager of Public Relations Charles L. Springer
Director of Fulfillment Services Bonnie Shepard
Assistant Customer Service Manager Deidra Henry
Customer Service Representative Sandy Apple, Monica Wheat
Word Processor Manager Lynda Wilson
Research Assistants Judy Brashear, Laurie Falk, Sharon Smith
Dispatch Janice Eastburn
Production Assistant Melba Smith

For PCM Advertising and Marketing Office information, see Page 90.

PCM — The Personal Computing Magazine for Tandy® Computer Users is published every month of the year by FALSOFT, INC., 9529 U.S. Highway 42, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Personal Computing Magazine for Tandy® Computer Users** and the **PCM** logotypes are registered ® trademarks of FALSOFT, Inc.

Second class postage paid Prospect, KY, and additional offices. USPS N 713-470 (ISSN 0747-0460). POSTMASTER: Send address changes to PCM, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.

Entire contents copyright © 1985 by FALSOFT, Inc. **PCM — The Personal Computing Magazine for Tandy® Computer Users** is intended for the private use and pleasure of its subscribers and purchasers and reproduction by any means is prohibited. Use of information herein is for the single end use of purchasers and any other use is expressly prohibited. All programs herein are distributed in an "as is" basis, without warranty of any kind whatsoever.

TRS-80, Model 100, and Portable Computer are registered ® trademarks of the Tandy Corp. CompuServe is a registered ® trademark of CompuServe Inc.

Subscriptions to **PCM — The Personal Computing Magazine for Tandy® Computer Users** are $28 per year in the United States. Canadian rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only.

Limited back issues are available. Please see notice for issues which are in print and costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

PCM

Contents

Features

2000 Colors/ *John B. Harrell III* 10
Selecting colors for your Tandy 2000

Barden's Buffer/ *William Barden, Jr.* 18
Sequential file techniques

The Gallery/ *Wayne Sanders* 28
Our logo on your computer

Connecting to ViaNet/ *Danny Humphress* 34
A first look at Tandy's networking system

Subroutine City/ *Robert D. Covington* 41
Calling MS-DOS from BASIC

A Better Input Processor/ *Bill Qualls* 49
Data entry subroutines for Model 100/200 BASIC

Manipulating RAM Files in the Tandy 200/ *Carl Oppedahl* 57
Access Model 200 memory files from BASIC

MS-DOSsier/ *Danny Humphress* 63
Use device drivers to configure your MS-DOS system

Portable Multiple Choice Test Generator/ *Fred Scerbo* 66
Create professional-looking paper tests with ease

dBase Tutor/ *Danny Humphress* 71
Designing a mailing list program

Telecommunicating with Randy/ *Randy Graham* 74
Transmitting graphics

Departments

Advertisers Index 90
Back Issue Information 79
Bar Coded Listings 86
Letters to PCM 8
Lprint/ *Lawrence C. Falk* 6
Editor's Notes
New Products 85
The Rackseller 90

Reviews

XTREE/ *Executive Systems Inc./1000, 1200, 2000* 77
TuneSmith/ *Blackhawk Data Corporation/1000, 1200* 78
DATA/SORT/ *Tandy/100, 200* 79
Synonym Finder/ *Writing Consultants/1000, 1200, 2000* 81
Sargon III/ *Hayden Software Company/1000, 1200* 82
Men-U-Tility & ByteFyter/ *Custom Software/100* 83

Cover illustration by Fred Crawford copyright© 1985 by Falsoft, Inc.

Tandy computers offer everything . . . almost

Check out the software that completes their systems —*filePro*®

Tandy MS-DOS Software Comparison Chart

	pfs® File/Report	dBase™ II	*filePro* 16 Smallware
GENERAL CHARACTERISTICS:			
Menu driven	yes	no	YES ✓
Allows user to create integrated business systems	no	programmer required	YES ✓
Developed systems and data can be moved to multi-user environments	no	no	YES ✓
Professional support available from the software's authors	no	no	YES ✓
PRICE	$265	$595	$495
CAPACITIES:			
Fields per record	100	32	999 ✓
Characters per record	1679	1000	4608 ✓
Records per file	1300	65535	16,000,000 ✓
Indexes per file	1	7	12 ✓
Number of digits per numeric field	20	10	24 ✓
Number of files usable concurrently	1	2	10 ✓
Files span multiple drives	no	no	up to 8 ✓
FEATURES:			
Full-screen facility for creating custom screen layouts	yes	no	YES ✓
Full-screen facility for creating custom report layouts	no	no	YES ✓
Built-in field types (error checking)	no	3	12 ✓
User-defined field types	no	programmer required	200 ✓
Conditional math	no	programmer required	YES ✓
User-defined menus	no	programmer required	YES ✓
Change file layout without losing existing data	possible	possible	automatic ✓
Data protection	no	programmer required	YES ✓
Password security	no	programmer required	YES ✓

The new Tandy computers can really speed things up. If you don't let obsolete software bog you down.

Try *filePro* 16, the latest generation of data base software from the authors of Profile®.

Choose it for its power (check our capacities); its ease-of-use (check our features); and its unique ability to grow as your needs grow, and even move the systems you create to multi-user computers.

filePro is available for the Tandy 1000, 1200 and 2000 computers (hard disk required). Available directly from us or through your Radio Shack store's Express Order Software service.

To order, call (800) 847-4740.

For customer service or when dialing in New York State, call **(212) 398-9290.** Ask for Smallware sales. Visa and MasterCard accepted.

Also available for other computers, including IBM PC, IBM PC-XT, and NCR Decision Mate.

The Small Computer Company, Inc.

✓240 230 West 41st Street, Suite 1200, New York, New York 10036

Smallware and *filePro* are registered trademarks of The Small Computer Company, Inc. dBase II is a trademark of Ashton-Tate. pfs File/Report is a registered trademark of Software Publishing. Profile is a registered trademark of Tandy Corporation.

Test Driveable Software —

Best Marketing Idea to Come Down the Pike

To my mind, anyway, one of the biggest issues in the world of software selection is that there is really very little opportunity to see just how well or poorly a product is matched to your individual needs.

While at the recent Summer Consumer Electronics Show in Chicago, I had a very interesting chat with the marketing director of a firm which publishes what I consider to be a very fine integrated package. He was bemoaning the fact that sales personnel in almost every computer store are woefully unable to really assist clients in selecting software for their particular needs.

"A guy learns to run *Lotus 1-2-3* and he thinks he's a computer whiz," was the lament. "With no disrespect to Lotus, one of the reasons they have so many sales is simply because their program is all that half the sales people in the world know how to run.

"It sure would be nice if people could take a product to their home or office, try it out, and then see if it is *really* what they want," he added. "Of course, with the tremendous cost of these programs — and the piracy problem — that's just not possible."

This problem becomes more acute when dealing with mail-order programs. And, since quite a number of programs *are* sold through mail order, there has to be some other way.

Add to this the way most firms address this problem — with "demonstration disks" that either do not do everything the program does or just flash up sample screens — and you do have an interesting issue.

Even with something as expensive as an automobile, you get to take a "test drive." Why can't it be possible to "test drive" software — the full version, not some watered down "demo?"

Now you can.

A company called Software Concepts, Inc. has a new scheme that allows you to have a full-blown demonstration of one of its new programs — a computerized Atlas. If you like what you see, you call them, they give you a code to "unlock" the disk, and charge you for it. If you don't like it, you simply reformat the disk and use it as a blank.

There are a couple of ways this works. One of them is to send Software Concepts $5 and they will send you the disk and the documentation. They figure $4 is the cost of the disk, $1 is the shipping charge. So, at the worst, you get a disk for $5.

The disk contains the entire program. You are allowed three 40-minute sessions with it. During those sessions, you can do anything you wish — this is a "full blown" version of the Atlas.

I have tried this out and it works well. This is not the appropriate place for a review of a program, so we will leave that for later. But I am highly encouraged by the concept and believe it represents an innovative and skillful way to market software. Software Concepts' idea means *you* have sufficient time, your own time, to make your own decision. I hope other firms will look at this and other innovative means of marketing in the future.

I am pleased to be able to report that PCM is growing by leaps and bounds and that, obviously, we are expanding again. We certainly got a lot of positive comments on last month's issue — the first time we had "perfect bound" our magazine.

Perfect binding is what the "square back" look is called. It gives us a little more flexibility and means that we can have a nicer looking magazine as well. I hope you like it.

I don't mean to make this last, but the reaction to PCMfest has been excellent thus far and I hope you will make plans to be with us in Princeton, N.J. October 11-13. There's a ticket order form in this month's issue and you can save a few bucks by advance ordering your tickets.

Too, I would suggest you make your reservations now for hotel space at the Hyatt Regency-Princeton if you will need to stay over with us (hope you do!) because sometimes the convention hotel sells out early. If so, you miss the guaranteed room rates and have to drive back and forth from someplace else.

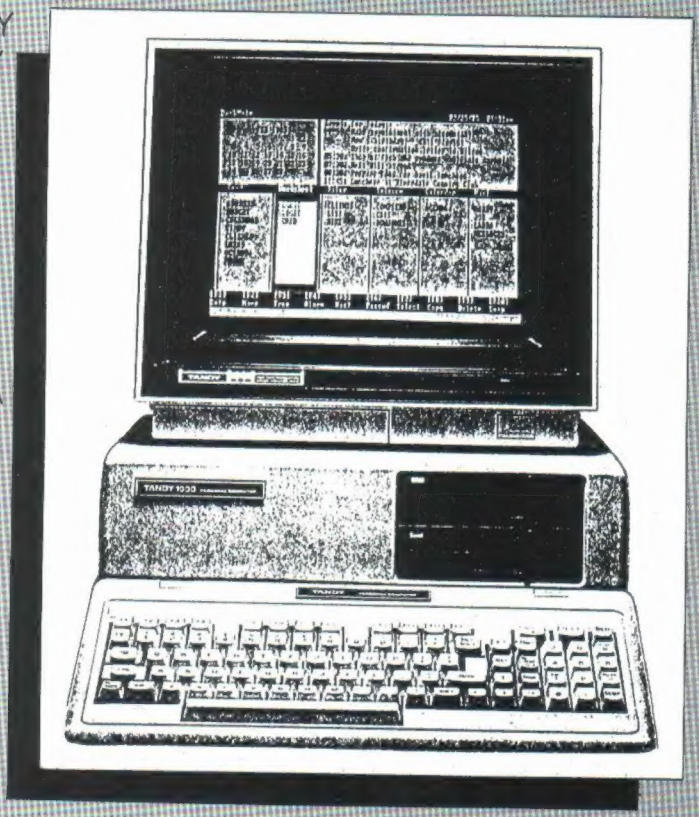
One last note: we can't guarantee that Brooke Shields will be at PCMfest this year, even though she lives in town and attends Princeton University. She was at the hotel for the weekend when RAINBOWfest was at the same place last year.

— Lonnie Falk

TANDY 1000
INTERNAL HARD DRIVES

- NO ADDED MEMORY BOARD NECESSARY
- BOOTS OFF HARD DRIVE
- TERMS—
  MONEY ORDER
  CASHIER'S CHECK
  *C.O.D.
  *MASTERCARD/VISA
  5% FEE

*PLUS SHIPPING & HANDLING



- COMPLETE KIT—NOTHING ELSE TO BUY
- INSTALLS IN MINUTES WITH ONLY A SCREWDRIVER
- INTERNAL KITS AVAILABLE FOR*:
  TELEVIDEO
  TANDY 1000
  SANYO
  COMPAQ
  LEADING EDGE
  PANASONIC
  CORONA
  IBM PC
  NOVA
  AT&T, NCR & ITT

10 MEG	20 MEG	32 MEG
$895.00	$1195.00	$1395.00

HARD DRIVE & CONTROLLER ARE 100% IBM COMPATIBLE. WE USE ONLY QUALITY DRIVES & WESTERN DIGITAL CONTROLLERS FOR THE HIGHEST QUALITY. WE BACK THE FULL SYSTEM WITH A 1 YEAR WARRANTY. THE SYSTEM COMES COMPLETE WITH ALL CABLES & MOUNTING HARDWARE. REQUIRES NO ADDED MEMORY BOARD TO OPERATE. 20 & 32 MEGABYTE INTERNAL HARD DISKS ALSO AVAILABLE. COMPLETE INFO AVAILABLE UPON REQUEST.

RTX TECHNOLOGIES, INC.
P.O. BOX 50398
NASHVILLE, TENN. 37209
(615) 373-1251

*SAME PRICE AS TANDY
*ALL MS™-DOS COMPUTERS

T1K SATISFACTION

Editor:

I am a fairly new user of a Tandy 1000, but I am not new to computing as I have had a Color Computer since the days when THE RAINBOW was only 42 pages long. When I needed to expand, I naturally looked to Tandy. I have not been disappointed since the 1000 will do all I expected of it, however there are one or two things I would like to clarify, and that is why I am writing to you, in the hope that someone in "Tandyland" can help me.

The colorful brochure on the 1000 published by Tandy mentions several features of the 1000 that are not covered in any documentation I have seen (these are all in the small print at the bottom of the last page); they are: user definable character set, double high characters and smooth scrolling and split screen. Can anyone throw any light on them?

I have a DMP 200 connected to my 1000. It worked flawlessly when coupled to my Color Computer, yet when driven by the 1000 I occasionally get the random insertion of an extra character, always the duplicate of the one preceding it. This happens regardless of the software running at the time. Has anyone else seen this, or is it a bug in my particular machine?

Finally, the joysticks under BASIC. When I try the short program listed on page 313 of the *BASIC Reference Manual* the values displayed on the screen do not appear in a logical sequence. As near as I can tell they should start at 1 (or 0) for the far left position and increase progressively to 255 (I think) at the far right (this is for `STICK(2)`). Instead they increase from 4 and then rapidly oscillate between a low number and a number over 200; they even do this with no joystick plugged in. Am I doing something wrong, or is BASIC incomplete for this function, or is my A-D converter malfunctioning? It is not my joysticks, since I get the same result with brand new deluxe joysticks and with the mouse!

Thank you for any assistance you can give me.

C.F. Thompson
Nelson, British Columbia

Editor's Note: The Tandy 1000 does indeed have a video mode which allows you to define your own character set (a standard feature on the IBM PC). Like IBM, though, Tandy does not supply software to allow the average user to redefine the characters. We have seen an IBM PC program in the public domain for this function, but do not have one available to print. If we come across one, you'll see it in these pages.

Unlike the Tandy 2000's version of MS-DOS, the 1000's DOS does not have a built-in facility for smooth scrolling. The machine is capable of it, but once again, it is a matter of software.

Concerning your printer problems, it is possible that your printer is at fault. When the printer is connected to your Color Computer, you are using the serial (RS-232) interface instead of the parallel interface you use with the Tandy 1000. Since the parallel interface is different circuitry, there could be a problem there and the machine would work perfectly when connected serially. Then again, the problem could be as simple as a bad cable. Before taking your printer to be serviced, we suggest borrowing another cable and trying it.

We have experienced the same problems here when using the joysticks in BASIC. When used with other programs, they seem to work fine. It looks like BASIC is the problem. The new version of BASIC, which will be released along with the new version of MS-DOS, should solve this problem. It's expected to be available this month.

PCM ON COURSE

Editor:

The `FBYTES.CO` article by Ron Balonis (May 1985) clearly shows that PCM is going in the right direction. Not only is `FBYTES.CO` a useful program in itself, but the accompaning article is also informative on the operation of the machine language program. The source code and description of the operation of the program in the article were particularly helpful.

Also on the subject of assembly language, I have typed in the listing that appeared in the April 1985 issue of PCM. I have not been able to find all of my typos. Is it possible to purchase a copy on cassette or bar code from PCM or the author?

Bennett D. Shulman
Lansing, MI

Editor's Note: `FBYTES.BA` *was printed in bar code in that issue. It is a BASIC program that generates the machine language program,* `FBYTES.CO`. *The assembly language listing was provided mainly to explain how the program worked and does not have to be keyed in if you use* `FBYTES.BA`.

HITCHING TO A STAR

Editor:

I've just received my second issue of PCM, and I must say that I am pleased with what I see. I recently purchased a Tandy 1000, and I am having a problem interfacing my Star Micronics Gemini

10X printer to it, and I'm hoping that you can give me some help.

The printer functions properly when printing text, but I'm having difficulty printing graphics. The GRAPHICS.COM file that comes with the Tandy 1000 will not drive the Gemini, since it is made for Tandy's CGP-220 Ink Jet Printer, but I pulled out my trusty IBM PC-DOS disk and found that the GRAPHICS.COM file that IBM supplies will drive a Gemini, sort of. The Graphics image is printed (using "SHIFT-PRINT"), but the printer inserts too long of a line feed between each printed line. I've connected my Gemini to a friend's IBM PC, and it behaves the same way.

Have any other Gemini owners encountered this problem? Is there a way to "fix" the PC-DOS GRAPHICS.COM file so the printer will print graphics correctly?

Any help would be appreciated very much. Thank You!

Chris Olonzo
Millington, NJ

Editor's Note: You might try changing a dip switch on your printer. There is usually a switch marked something like CR/LF which enables or disables the automatic line feed after carriage returns. Incidentally, the new version of MS-DOS for the Tandy 1000 which should be out later this month has a utility for printing black & white images on a number of Tandy printers. That will not help you with your Gemini, though.

USER GROUPS WANTED

Editor:

I have a TRS-80 and am an avid Tandy user. I am interested in contacting other Tandy users and User Groups. Do you have a back issue that lists the addresses of your current user groups? If so, I would be interested in receiving that issue or information on a contact person.

Rod Aldrich
Marquette, MI

Editor's Note: We have not printed lists of user groups, but it's a great idea!

Attention Tandy user groups! Send us your address, telephone number, a contact name, and a description of your group. We'll periodically print a list of users groups.

HIS WORD

Editor:

The recent review of *MY WORD!*, our word processor for the Tandy 1000/1200/2000 and all IBM PC-compatible computers, contains two inaccuracies.

First, the reviewer mentioned that he couldn't determine how to set the right margin until he stumbled across a method that appeared to work: depressing the ESC key.

In fact, instructions for setting the margin (in two different ways) appear in the manual. There is an entry in the second page of the table of contents of the manual which may have been overlooked by the reviewer. The entry reads: "Set right margin (≙OR) ..... 36."

On Page 36, both methods of setting the right margin are described; the alternate method of setting the right margin used by the reviewer is highlighted in boldface print.

The second error is of less consequence. The reviewer mentions that *MY WORD!* appears to emulate *WordStar*. He also states that this fact, if true, is not mentioned in the manual.

However, the introduction to the manual (see Page 9) contains the following sentence: "In case you haven't already guessed, the package I'm emulating is *WordStar*."

WordStar also appears in the index to the manual, and is cross-referenced to Page 9. The manual also contains several places where *WordStar* is acknowledged to be a trademark of MicroPro International, one of which is also on Page 9.

Despite these inaccuracies, we were happy with the content of the review. It was apparent that the reviewer was pleased with the quality and value of our package, especially in light of our price: just $35. We are truly grateful for the reviewer's comments and his recommendation of *MY WORD!*

Your readers should also note our guarantee. Our package may be returned for a full refund of the purchase price within 30 days.

Thank you again for your courtesy in sending a copy of the review to us. Your publication certainly combines high ethical standards with editorial quality, something all too rare today.

Bruce W. Tonkin
President, T.N.T. Software
Round Lake, IL

MOVIN' ON UP

Editor:

My wife is still using the Color Computer for word processing, but I have changed everything over to the Tandy 1000.

I have purchased *Lotus 1-2-3, dBASE II*, and the DAC Software accounting programs, so you can see the articles in PCM will be helpful.

Dollar for dollar I don't see how the 1000 could be beat. I am sure it will not be long before PCM is the size of THE RAINBOW. I just hope you will keep PCM more for business applications and leave the games and graphics for the Rainbow.

Bud Mott
Cushing, OK

NEW TANDY 200 24K EXPANSION MODULES $149
TWO EXPANSION MODULES FOR $275

8K MEMORY MODULE $35
For your Model 100 & NEC PC-8201. Field proven (over 20,000 in use), factory tested and fully supported.

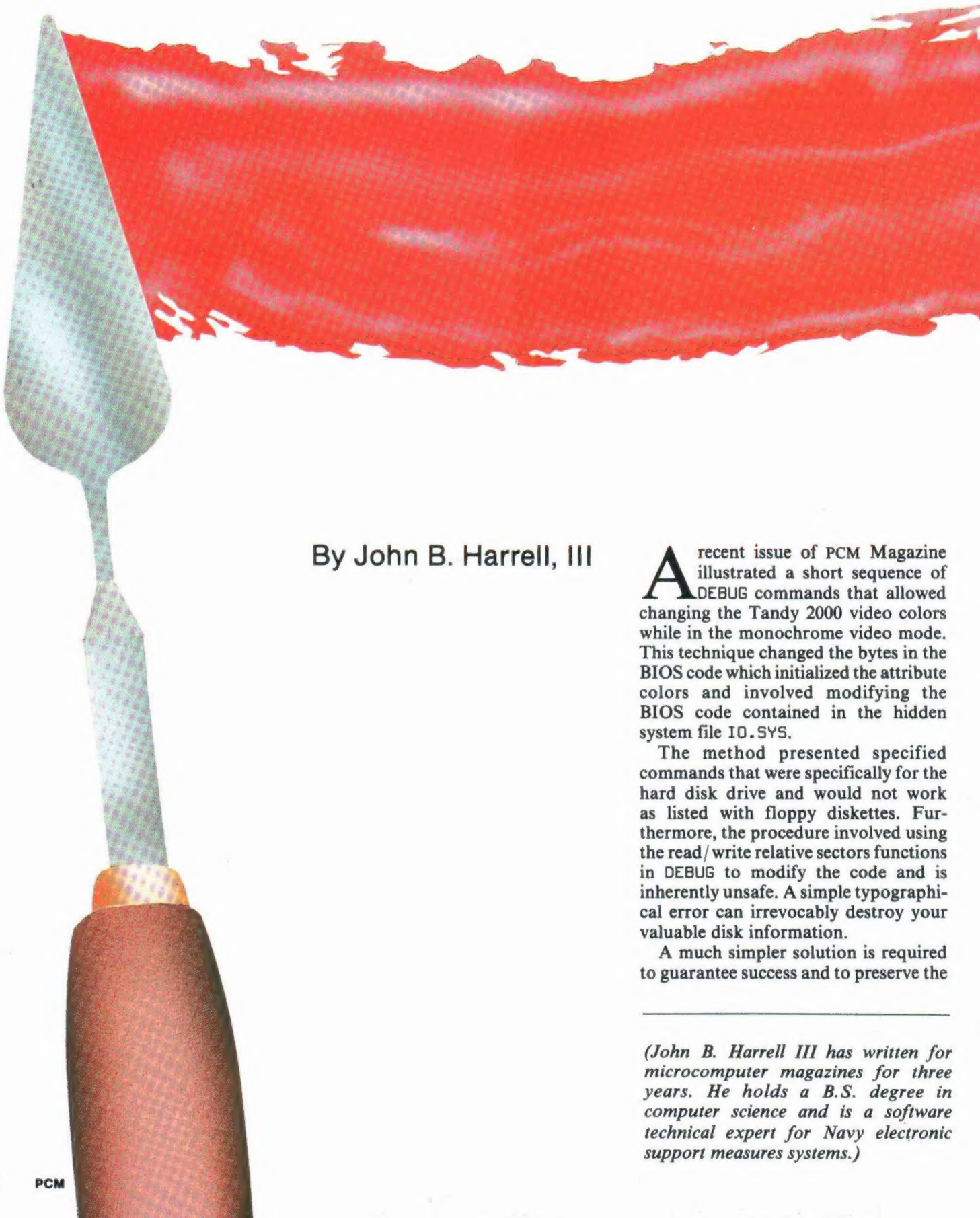
Just for your NEC PC-8201
SIDECAR 32K RAM CARTRIDGE $199
Expandable to 128K/Plugs into side of the NEC PC-8201
Each additional 32K RAM bank $100, installed.

Portable Thermal Printer
TTXpress MODEL 1280 $170
Battery operated, 2.2 lbs., 8½" paper, 40 cps., parallel.

FREE SHIPPING, HANDELING & INS. FOR CONT. USA.
Orders shipped UPS next business day. 30-day satisfaction, money back guarantee. We accept Visa, M/C & Amer. Exp. Checks held 18 days.

PURPLE COMPUTING
420 Constitution Ave.
Camarillo, CA 93010

(800) 732-5012 **(805) 987-4788** (in Calif.)



By John B. Harrell, III

A recent issue of PCM Magazine illustrated a short sequence of DEBUG commands that allowed changing the Tandy 2000 video colors while in the monochrome video mode. This technique changed the bytes in the BIOS code which initialized the attribute colors and involved modifying the BIOS code contained in the hidden system file IO.SYS.

The method presented specified commands that were specifically for the hard disk drive and would not work as listed with floppy diskettes. Furthermore, the procedure involved using the read/write relative sectors functions in DEBUG to modify the code and is inherently unsafe. A simple typographical error can irrevocably destroy your valuable disk information.

A much simpler solution is required to guarantee success and to preserve the

---

(John B. Harrell III has written for microcomputer magazines for three years. He holds a B.S. degree in computer science and is a software technical expert for Navy electronic support measures systems.)

Selecting Colors for your Tandy 2000

disk integrity. This article will provide a simple program which allows you to tailor the four color selections to your taste and then writes a simple program file that you can execute from any MS-DOS command prompt to set these colors.

The resultant program (I'll refer to it as COLOR.COM for the time being) will make the appropriate changes in the BIOS code which is resident in memory, achieving the same effect as the previous method of modifying the disk-resident code. Because the actual BIOS code is modified, the net result is identical to modifying the disk code and rebooting the system. The biggest advantage of this method is that you have not chanced losing all the information on the disk.

SETCOLOR — The Program

SETCOLOR is very easy to use — simply enter the program contained in Listing 1 using the BASIC interpreter. Enter the numbers contained in the DATA statements carefully as this is the heart of the color selection program. You will need a copy of the *MS-DOS Operating System Version 02.11.XX*. COLOR.COM will not operate correctly with the older version of MS-DOS.

Now, let's look at what SETCOLOR really does (all comments refer to Listing 1). First, SETCOLOR sets the video mode with color burst turned off (Line 30) and selects the black-and-white mode with the colors contained in the BIOS code. Lines 50 through 100 extract the color codes contained in the resident BIOS code and insure that the screen colors are set correctly by sending them to the appropriate port registers using BASIC's OUT command.

Lines 110 through 200 write the color selection prompts on the video screen using BASIC's COLOR statement to select the foreground and background attributes of the prompts. The background color (i.e., the color surrounding each of the letters) of each of these prompts matches the color that will be changed by altering this selection. The final prompt displays a menu of allowable entries using the "blink" mode.

The next segment of code from lines 210 through 320 accepts your input for each selection. First, a prompt is displayed using attributes identical to the selection that it matches *with* the text blinking. Next, the default color and corresponding port address are passed to the subroutine which accepts your input (GOSUB 510). The last statement of each group saves the selected color for later use in constructing the actual program.

The lines from 330 to 490 accept the filename of the COM file to be constructed with the selected color information. Once the file has been opened, COLOR.COM is constructed by reading the DATA statements and writing the information to the named file a byte at a time.

The last part of this program segment adds the selected color information to the end of this short program. Figure 1 contains a symbolic assembly language listing of the code contained in these DATA statements, with comments suitable for explaining what is accomplished.

Lines 510 through 570 accept the input for color selections. Colors are specified by hexadecimal digits (see Figure 2 for an index of colors and their numerical equivalents), and the alphabetic characters may be typed in either upper- or lowercase. If the character is a valid hexadecimal digit, it is converted to binary and output to the port address passed as a parameter to this subroutine.

Explanation of the B&W Color Mode

During normal color graphics mode of operation on the Tandy 2000, the video controller extracts display data from the RAM beginning at segment address E000H. The video graphics RAM consists of three planes (or banks) of 32,000 bytes of memory and setting any bit in any of the planes will set the corresponding pixel on the video screen.

Colors are displayed by setting one or more bits in a corresponding pixel location (bits in different video planes), which then select the color from the palette location corresponding to the binary pattern stored in the planes.

All three planes of screen data are

Figure 1
COLOR.COM BIOS Patch Routine

Generated Code	Instruction	Comments
31 C9	XOR CX,CX	;CLEAR CX REG TO USE IN ES
8E C1	MOV ES,CX	; AS THE SEGMENT FOR BIOS
BB 29 01	MOV BX,0129	;POINT TO THE COLOR VALUES
8A 07	MOV AL,[BX]	;GET FIRST COLOR VALUE
26	ES:	;SEGMENT OVERRIDE
A2 F3 1B	MOV [1BF3],AL	;STORE NORMAL BKGD COLOR
8A 47 01	MOV AL,[BX+1]	;GET SECOND COLOR VALUE
26	ES:	;SEGMENT OVERRIDE
A2 F9 1B	MOV [1BF9],AL	;STORE NORMAL FGD COLOR
8A 47 02	MOV AL,[BX+2]	;GET THIRD COLOR VALUE
26	ES:	;SEGMENT OVERRIDE
A2 FF 1B	MOV [1BFF],AL	;STORE HILITE BKGD COLOR
8A 47 03	MOV AL,[BX+3]	;GET FOURTH COLOR VALUE
26	ES:	;SEGMENT OVERRIDE
A2 05 1C	MOV [1C05],AL	;STORE HILITE FGD COLOR
BB 02 00	MOV AX,0002	;SET SERVICE/FUNCTION CODE
CD 10	INT 10	;SET SCREEN MODE BIOS CALL
CD 20	INT 20	;RETURN TO MS-DOS
WW XX YY ZZ	DB WW,XX,YY,ZZ	;COLOR BYTE VALUES

Note: This code sequence is designed for use with DEBUG and not the macro assembler. All numbers are in hexadecimal.

mapped into the same RAM address space. Selection of any one of these planes is controlled by the bit pattern output to I/O port 01A0H (refer to the *Hardware Reference Manual* for a complete description of the bit map associated with this port). This bit pattern also controls the selection of the graphics RAM for display or the selection of the monochrome RAM.

The monochrome RAM is not fixed at a particular segment address as is the graphics RAM. During initialization of the computer system, this RAM area is established as the last 5K bytes of your addressable memory space. For example, 3EC0H is the segment address for a 256K Tandy 2000 and 7EC0H is the segment address for a 512K RAM system.

Unlike the graphics RAM, the monochrome RAM is organized as a byte of character information followed by a byte containing that character's attributes. The exact description of the bits contained in this byte may be found in the *Hardware Reference Manual*. They specify that the byte preceding it may be displayed in normal or highlight mode and normal or reverse video. Other bits may specify blinking or underlined special features. Attributes may be combined in almost any selection.

Using the normal/highlight and normal/reverse video modes gives us the four different colors that may be displayed on the video screen in the black and white mode. The colors selected by these attributes are determined by the selector byte output to the port addresses described in Table 1. This is typically performed by MS-DOS during initialization and each time the video screen is restored to the black and white mode.

The small COM program generated by SETCOLOR (COLOR.COM) will set each of the colors you select into the appropriate locations in the BIOS code. After the BIOS code has been patched, COLOR.COM issues a BIOS interrupt request

Table 1

Port Address	Function
0198H	Normal background color typically set to black for standard MS-DOS
019AH	Normal foreground color typically set to white for MS-DOS
019CH	Highlight mode background color typically set to black
019EH	Highlight mode foreground color typically set to bright white

Figure 2
Tandy 2000 Color Selection Table

Number	Color Meaning	Number	Color Meaning
0	Black	8	Black
1	Blue	9	Bright Blue
2	Green	A	Bright Green
3	Cyan	B	Bright Cyan
4	Red	C	Bright Red
5	Magenta	D	Bright Magenta
6	Yellow	E	Bright Yellow
7	White	F	Bright White

Prices Change Every Day. Please Call 1-800-343-8841 For Lower Prices.

PRICE BREAKTHROUGH

Prices Change Every Day. Please Call 1-800-343-8841 For Lower Prices.

DEALER INQUIRIES INVITED

MEGADISK™ HARD DISK DRIVE SYSTEMS

TOLL FREE ORDERING 1-800-343-8841

For the { **IBM/PC**, Tandy 1000, **TRS/80 Models I/III/IV/4P**, Compaq, Eagle, Sanyo, Tava, PC Workalikes, **Color Computers**, Heath/Zenith, Max/80 Complete with Hardware, Cables, Software and Quikfit Installation

Wow! New Low Prices

5 Megabytes Internal Mount	starting at	$ 299.95
10 Megabytes Internal Mount	starting at	544.95
20 Megabytes Internal Mount	starting at	799.95
5 Megabytes External System	starting at	499.95
10 Megabytes External System	starting at	699.95
20 Megabytes External System	starting at	999.95
Streaming Tape Backup System — Internal Or External	starting at	499.95

DOS Systems Available: { IBM/Heath — DOS, 1.0, 2.0, 2.1, 3.0, or later; TRS/80-LDOS, TRSDOS 6.x, Newdos/80, Dosplus 3.5, CP/M, COCO DOS, Max/80 LDOS, OS9

FULLY WARRANTIED — PARTS AND LABOR — 24 HOUR SERVICE — CALL TOLL FREE — 1-800-343-8841

FLOPPY DISK DRIVES, POWER SUPPLIES AND CABINETS

Our Disk Drives are UL approved — Our Floppy Drive Cabinets and Power Supplies are Underwriters Laboratory Listed and have passed the required Federal Communications Part 15 Section B-EMI/RFI tests.

Warranty on all disk drives is one full year parts and labor. Warranty on floppy disk drive power supplies is five (5) years. In warranty or out of warranty service is 24 hour turn-a-round on all disk drives and power supplies.

Tandon — Full Height Drives		
100-1	Single Sided 40 tk Bare	$125.00
	In Case with Power Supply	166.95
	Dual Drives in One Cabinet	326.95
100-2	Dual Sided 40 tk Bare	162.00
	In Case with Power Supply	206.95
	Dual Drives in One Cabinet	381.95
	Half High Drives	
65-1	Single Sided 40 tk Bare	122.00
	In Case with Power Supply	156.95
	Dual Drives in One Cabinet	306.95
65-2	**Dual Sided 40 tk Bare**	**142.00**
	In Case with Power Supply	**176.95**
	Dual Drives in One Cabinet	**346.95**
65-4	Dual Sided 80 tk Bare	162.00
	In Case with Power Supply	196.95
	Dual Drives in One Cabinet	356.95
TEAC Half High Drives		
55A	Single Sided 40 tk Bare	125.00
	In Case with Power Supply	166.95
	Dual Drives in One Cabinet	316.95
55B	Dual Sided 40 tk Bare	157.00
	In Case with Power Supply	196.95
	Dual Drives in One Cabinet	356.95
55F	Dual Sided 80 tk Bare	177.00
	In Case with Power Supply	216.95
	Dual Drives in One Cabinet	386.95
Apple/Franklin Disk Drives		
	35/40 Track in Case with Cable and Software	159.95
	Controller Card for Two Disk Drives	46.95
	Combination Price for Disk Drive and Controller	196.95
Commodore Disk Drives		236.95
Power Supplies and Cabinets 5¼, and Hard Drive Systems		starting at 42.00

COLOR COMPUTER DISK DRIVE SYSTEMS AND ADD IN PRODUCTS

40 Track Single Head Drive with Case, Power Supply, Cable Controller, Instruction Booklet, Diskettes	$239.95
Above with Dual Drives in One Cabinet	374.95
40 Track Dual Head with Case, Power Supply, Cable, Controller, Instruction Booklet, Diskettes	**259.95**
Above with Dual Drives in One Cabinet	**399.95**
128 Memory Upgrade Kit	129.95
Dual DOS Switch	19.95
With Second DOS System — JDOS, RSDOS, Micro DOS, and Booklet	69.95

MODEMS

Volksmodem 300 Baud	$ 69.95
Signalman Mark X Autodial	123.95
Mark XII 1200/300 Baud Autodial	284.95

ALL IN-STOCK ITEMS SHIPPED WITHIN 24 HOURS. SAME DAY SHIPPING PROVIDED BY REQUEST WITHOUT ANY EXTRA HANDLING CHARGES.

********** MEGADISK SPECIAL **********

With a 10 megabyte system, you can purchase a complete IBM or PC compatible host, software and cable for only $15.00!!
P.S. Ask about the MEGAPLEXER.

IBM - PC/XT WORKALIKE

All of our computers have: 8 slot motherboard, 640K, monochrome adapter, parallel printer port, 130 watt power supply, free software — all with our full warranty.

/PC- 2 Floppy Drives - Monochrome Monitor, MS-DOS	$1,595.00
Color system resolution - 720h x 240v, MS-DOS	$1,995.00
5meg/**XT**tra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	$1,745.00
Color system resolution - 720h x 240v, MS-DOS	2,145.00
10meg/**XT**tra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	2,095.00
XTtra - Color system resolution - 720h x 240v, MS-DOS	2,495.00
20meg/**XT**tra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	2395.00
XTtra - Color system, MS-DOS	2,795.00
Internal Tape Backup For Any Of Above systems Add	$449.95

PRINTERS

Dot Matrix		
Citizen		$ Call
Star Micronics		
Gemini 120 CPS Near letter quality	starting at	$259.95
Delta 10/15 160 CPS	starting at	424.95
Radix 10/15 200 CPS	starting at	649.95
Panasonic 1090		249.95
Daisy Wheel		
Silver Reed 440 80 Column 12 CPS		315.95
550 132 Column 19 CPS		439.95
770 132 Column 36 CPS		895.00
Olympia 132 Column 14 CPS with Form and Tractor		399.95
Apple/Franklin Printer Interface w/Graphics and Cable		84.95
Printer Cables	starting at	19.95
Printer Paper — Microperf Edge 1000 Sheets		16.95

ELECTRICAL

Surge Protectors — Line Filters — SL Waber — 6 Outlets with Switch	$ 39.95
Uninterruptable Power Supplies	399.95

MISCELLANEOUS

Diskettes in 10 Pack	**from**	**$ 10.95**
Twoprint Switches	from	99.95
Disk Drive Cables	from	16.00
Maintenance Cleaning Kits		12.00
Parallel Printer Buffers 8K		149.95
Floppy Disk Drive Cables		
1 Drive		16.00
2 Drives		18.95
Heath/Zenith 2 Drive Cables — Shielded		24.95

SOFTWARE SUPPORT, INC.

1 Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425

Hours: Mon. thru Fri. 9:30 am to 5:30 pm (E.S.T.) Sat. 10 am to 3:30 pm

SERVICE POLICY — Our Professional Technical Staff Is Available To Assist You Monday Through Saturday.
WARRANTIES — Up To One Full Year Parts And Labor. Floppy Disk Drive Power Supplies — Five (5) Years.
SERVICE — 24 Hour Turn-A-Round On All In-Stock Parts. Dealer Inquiries Invited. Call 617-872-9090

Please Call For Shipping, Handling And Insurance.

Toll Free 1-800-343-8841

Please Call For Our Latest Price Saving Specials.

Not Responsible for Typographical Errors. Prices and Specifications May Change Without Notice.

to set video mode 2. This mode establishes an 80-by-25 monochrome character display and causes the subroutine just patched to be executed. This sets the new colors for your monitor.

Conclusion

The Tandy 2000 color display is a super monitor; being able to use the full color potential in all modes really lets you get the most from your computer. SETCOLOR provides a safe and easy method to change the colors and prevents you from potentially destroying your valuable disks.

This method of changing colors will work with most software for the Tandy 2000. The exceptions to this rule fall into two categories: those programs that must work in the color graphics mode, such as Microsoft's *Word*, and those programs that set the monochrome ports themselves, such as *Lotus 1-2-3* and *Open Access*.

Other software, such as *WordPerfect*, can be installed in the monochrome mode running in the full-color mode and allowing full use of attributes, such as on-screen underlining. Software which automatically selects the monochrome text modes (for example, *MultiMate*) will now run in full color.

Once you get familiar with DEBUG, the code sequence generated for the COM file can be entered directly using the assemble (A) command or enter (E) command. The code may then be written to a disk file using the write (W) command for later execution. You can easily change colors at any time. Enjoy your color Tandy 2000 to its fullest!

The listing:

```
10 DEFINT A-Z
20 SCREEN 3
30 SCREEN 0,0
40 KEY OFF
50 DEF SEG = 0
60 CLR.1 = PEEK(&H1BF3)
70 CLR.2 = PEEK(&H1BF9)
80 CLR.3 = PEEK(&H1BFF)
90 CLR.4 = PEEK(&H1C05)
100 OUT &H198,CLR.1: OUT &H19A,CLR.2: OUT &H19C,CLR.3: OUT &H19E,CLR.4
110 LOCATE 6,20
120 PRINT " The background of this text is color 1 ";
130 LOCATE 8,20: COLOR 0,7
140 PRINT " The background of this text is color 2 ";
150 LOCATE 10,20: COLOR 15,0
160 PRINT " The background of this text is color 3 ";
170 LOCATE 12,20: COLOR 15,7
180 PRINT " The background of this text is color 4 ";
190 LOCATE 24,10: COLOR 25,7
200 PRINT " Select a color (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) or press ENTER ";
210 LOCATE 18,32: COLOR 18,0
220 PRINT " Select color 1 ";
230 CLR = CLR.1: PORT.REG = &H198: GOSUB 510: CLR.1 = CLR
240 LOCATE 18,32: COLOR 16,7
250 PRINT " Select color 2 ";
260 CLR = CLR.2: PORT.REG = &H19A: GOSUB 510: CLR.2 = CLR
270 LOCATE 18,32: COLOR 31,0
280 PRINT " Select color 3 ";
290 CLR = CLR.3: PORT.REG = &H19C: GOSUB 510: CLR.3 = CLR
300 LOCATE 18,32: COLOR 31,7
310 PRINT " Select color 4 ";
320 CLR = CLR.4: PORT.REG = &H19E: GOSUB 510: CLR.4 = CLR
330 COLOR 7,0
340 CLS
350 LOCATE 5,1: PRINT "Enter a filename of 1-8 characters (no extension) ";
360 LINE INPUT B$
370 LOCATE 7,1: PRINT "Creating color mods as ";B$;".COM"
380 OPEN "R",1,B$+".COM",1
390 FIELD 1, 1 AS C$
400 FOR I=1 TO 41
410 READ C
420 LSET C$=CHR$(C)
430 PUT 1
440 NEXT
```

```
450 LSET C$=CHR$(CLR.1): PUT 1
460 LSET C$=CHR$(CLR.2): PUT 1
470 LSET C$=CHR$(CLR.3): PUT 1
480 LSET C$=CHR$(CLR.4): PUT 1
490 CLOSE
500 END
510 A$ = ""
520 WHILE A$="": A$=INKEY$: WEND
530 IF ASC(A$)=13 THEN RETURN
540 IF A$ >= "0" AND A$ <= "9" THEN CLR=ASC(A$)-48: OUT PORT.REG,CLR: GOTO 510
550 A$=CHR$(ASC(A$) AND &HDF)
560 IF A$ >= "A" AND A$ <= "F" THEN CLR=ASC(A$)-55: OUT PORT.REG,CLR: GOTO 510
570 GOTO 510
580 DATA &H31, &HC9
590 DATA &H8E, &HC1
600 DATA &HBB, &H29, &H01
610 DATA &H8A, &H07
620 DATA &H26, &HA2, &HF3, &H1B
630 DATA &H8A, &H47, &H01
640 DATA &H26, &HA2, &HF9, &H1B
650 DATA &H8A, &H47, &H02
660 DATA &H26, &HA2, &HFF, &H1B
670 DATA &H8A, &H47, &H03
680 DATA &H26, &HA2, &H05, &H1C
690 DATA &HB8, &H02, &H00
700 DATA &HCD, &H10
710 DATA &HCD, &H20
```

PCM

Introducing...

The Largest Selection of High Quality **Battery-Powered** Peripherals

Take Your Pick . . . With You!

ThinPrint 80™

40 CPS Thermal Dot Matrix Printer with Hi Res Graphics. Silent. Built-in Paper Compartment. 5 lbs.

ThinWrite 80™

50 CPS Dot Matrix Impact Correspondence or Near Letter Quality. Prints 3-part Forms and P.C. Graphics. 6 lbs.

ThinType 80™

15 CPS Bi-Directional Letter Quality Impact Printer. Fully Formed Char. Internal A/C Charge/Adapter. 7 lbs. (5 lbs. optional)

ThinView 25™

80 Char. x 25 Line LCD Intelligent Display System. Optional Tactile Input. Bit Image Graphics. 2 lbs.

Built-in Rechargeable Batteries save you time and dollars in replacements.
Proven parallel and serial interfaces on M100, PX-8, 2C, HP-IL and many others.

Axonix

Call Toll Free **800-821-7093** or 801-582-9271
DEALER INQUIRIES INVITED

The Portable Peripherals People
417 Wakara Way, S.L.C., Utah 84108

Text processing power that no other program can equal.

Write ROM©

on Snap-in™ Cartridge $149.95

PCSG says "Send it back in 30 days for a full refund if you don't agree."

WRITE ROM is the definitive word processing extension for the Model 100. Less than two months after the Model 100 was announced Portable Computer Support Group introduced the very first text formatter for the Model 100. That program, called Write + was licensed to Tandy and is now in Radio Shack Computer Centers as Scripsit-100. Write + had many powerful features and most reviewers still say it is the best of the cassette based text formatters. But now eighteen months later PCSG has introduced WRITE ROM. Those who experience it have said "WRITE ROM literally doubles the text processing power of the Model 100."

WRITE ROM is what you would have expected PCSG, the software leader for the Model 100, to develop in the 18 months since Write + was brought to the market.

First of all WRITE ROM as its name implies is on a snap-in ROM. You simply take a quarter and open the little compartment on the back of your Model 100 and press in the ROM cartridge. It is as easy as an Atari game cartridge and can be snapped in and out instantly so that you can use other ROM programs whenever you wish.

WRITE ROM appears on the main menu just like one of your built in programs. It lets you do every formatting function you would expect like setting margins, centering, right justifying and having headers and footers. But it does them under function key control, with the clear and easy to learn and use techniques for which PCSG has become famous.

In keeping with PCSG's long standing reputation for superlatively simple yet comprehensive documentation, the manual is a model of lucidity.

WRITE ROM remembers your favorite format settings so that you can print a document without any set up, but you can change any formatting or printing parameters instantly with a function key.

WRITE ROM's 'pixel mapping' feature shows you an instant picture on the screen of how your printout will look on paper. Incidentally, PCSG introduced this feature on the Olivetti M-10 version of Write + over a year ago.

In all there are 44 separate features and functions that you can do with WRITE ROM, and some of these features are truly breakthroughs for the Model 100.

First, WRITE ROM lets you do search and replace, with function key ease of course. Any word or phrase in a document can be searched for and replaced with any other phrase where the search words appear.

Second, WRITE ROM lets you send any text (formatted or not) to any other computer over the phone with just a function key. What's more, it dials and handles sign on protocol automatically.

Third, WRITE ROM has a wonderful feature called 'Library' that gives your Model 100 power that you never thought it could have. Library lets you record favorite phrases, words, or commonly used expressions (sometimes called boilerplate). Any place you wish any library text to appear in your document you just type in a code. WRITE ROM automatically inserts the text just like a Xerox Memory Writer.

The library phrase is inserted as your document is being printed rather than as it is being typed, so this feature conserves memory in documents where a long phrase is used repetitively, since each occurrence of a library phrase in your document is indicated by a single code character.

This Library feature is so powerful these two pages could be devoted just to telling you about things it can do. For example, you can have names and addresses that you designate in one text file with a customer or supplier number. Or you can have inventory items with stock numbers.

In your document you simply type in the customer or stock number and that entry from the other file is automatically inserted in the document. Picture what you can do with that kind of capability.

Because WRITE ROM is written in machine code, it is blindingly fast. No one can claim faster operation.

Because it is on a ROM it uses virtually none of your precious RAM for its operation, and it does not interfere with other machine code programs in your RAM. It works with any printer, serial or parallel. At the touch of a function key you can find the size of a RAM file in bytes and in words (ideal for journalists and other writers who need to know how many words are in a piece). You can make a duplicate copy of a document file under a new name. You also can rename or delete (kill) any RAM file with function key ease.

This description only scratches the surface of this amazingly powerful piece of software. You can automatically insert the date or the time anywhere in your document; WRITE ROM senses when you are nearing the bottom of a page, and at your command will start a new paragraph on the next page.

Write + was the Model 100 pioneer in the use of 'dot commands' to allow control of such things as margins, centering, line spacing and other appearance related changes in the middle of a document. WRITE ROM goes a step further by making all the dot commands Wordstar compatible. This means that if you wish you can quite easily prepare a Wordstar compatible document. Then you can use features of WRITE ROM (such as pixel mapping) that Wordstar lacks, before uploading to your desktop.

A Mail Merge feature allows you to send the same document to every name on your mailing list, personalized for each recipient.

WRITE ROM enables you to do underlining, boldface and correspondence mode as well as any other font feature that your printer supports in a way that is so unique many users say "It is worth the price of the program just to have this one feature."

Here's how it works: When you want to underline you don't have to remember some complicated printer code. You just type Graph-U, and to end underline you just type Graph-U again. For boldface it's Graph-B and to end boldface it's Graph-B again. It's easy to remember and easy to do. WRITE ROM lets you record the codes from your printer's manual one time only and then just use these easy to remember signals any time you want to do a printer font feature.

WRITE ROM does so many things that other text formatters cannot do. For example you can not only double space but triple, quadruple or any other.

WRITE ROM allows you to use your TAB key in a document so that you can indent the first line for a paragraph easily or space rapidly over many tab stops.

WRITE ROM has another nice feature. It allows you to undent. This means that you can have paragraphs that have a first line that projects to the left of the remainder of the paragraph.

WRITE ROM allows you to not only center a word or phrase on a line but you can center copy vertically on a page as well.

WRITE ROM has a feature that is unique to any word processor on any computer. It is called FORM. FORM is an interactive mechanism that lets you create screen prompts so that you or someone else can answer them to fill out forms, or supply information like to a questionnaire or answer correspondence rapidly inserting personal answers into a form letter.

It works sort of backwards from Library or boilerplate. As you recall, with the Library feature you type a code into a document and when you print, that phrase or word or paragraph is picked up from the Library file and inserted into the printed document. With FORM when you print, anyplace where you had previously typed in a GRAPH T in a document, the printer will stop and you are shown a prompt on the screen. You can type in directly on the screen and when you press ESC, what you typed is sent to the printer formatted like the rest of the document.

What is really great is that you created those prompts that appeared on the screen. By the way, the prompts won't appear in the printed document unless you want them to, and you don't have to be connected to a printer, you can write your completed forms to RAM files if you wish.

Think of how you can use FORM. A doctor or nurse could use it for a patient's history with each question appearing on the screen. An insurance salesman could have his entire questionnaire, or a police department could do a complete arrest report. You can construct a series of prompts to answer correspondence, automatically inserting the answers into a generalized letter format for a given type of correspondence, like customer service. This feature lets you answer letters in a rapid fire fashion each one with its personalized responses.

Before WRITE ROM you had to be a programmer to create a series of prompts to answer questions or record information. Now it is as simple as typing Graph T.

There are many other examples of excellent programming evident in WRITE ROM. The line feed problem of the Model 100 is dealt with by the simple use of a function key. Files are selected by moving the wide bar cursor over the WRITE ROM menu.

PCSG makes the claim that WRITE ROM is the easiest, fastest and most feature rich text formatter for the Model 100, as well as being the only one on a Snap-in ROM. You can do more with WRITE ROM than anyone thought possible for the Model 100. We at PCSG are happy to offer WRITE ROM because it expands the Model 100 to a dimension of text processing you cannot equal on even larger computers.

If you are already a PCSG customer you know the impressive quality of PCSG craftsmanship. We brashly state that WRITE ROM is the best you can buy. But don't take our word for it. It is sold on a thirty day trial. If you aren't as excited as we are, return it within 30 days for a full refund. Priced at $149.95, on Snap-on ROM. Mastercard, Visa or COD.

PORTABLE
COMPUTER
SUPPORT
GROUP

11035 HARRY HINES BLVD. No. 207
DALLAS, TEXAS 75229

PCSG provides hotline software support for Model-100. Call 1-214-351-0564. Available now directly from Portable Computer Support Group. We endeavor to continue as The Leader in Software for the Model 100

. . . Confessions of an ASCII bigot

High Praise for Sequential Files

By William Barden, Jr.
PCM Contributing Editor

I'll admit it, I'm an ASCII bigot. I *hate* encoded files that won't print out or display in understandable text characters. I use *My Word!* (TNT Software) because it doesn't use encoding such as *WordStar* - everything is in nice, displayable characters. I've even taken to saving all my BASIC programs in ASCII. Why use ASCII? It's just so *darn* easy to process the text files in a variety of different ways. Let me give you an example . . .

I had contracted with a small publisher for two books, one on the Tandy 1000, 1200, and 2000 and the other on the IBM PC and (ugh!) PC*jr*. Along with generating the manuscripts for the books, I had also agreed to insert minor typesetting commands into the text.

It turned out that the publisher's interpretation of "minor" and my version of "minor," differed greatly, as I was to find out. I wrote up about 10 double-spaced manuscript pages for samples, and then, manuscript in hand, went in to see the publisher's representative.

(William Barden, Jr., is a master communicator in a field in which he is one of the few recognized experts — microcomputers. A prolific author of more than 27 books and handbooks on computers and computer programming, Bill also has authored several instructional software projects for Tandy/Radio Shack.)

After the usual pleasantries, we got down to the business of how to edit the word processing files to the typesetting format.

"The first thing you've got to insert is a special code that tells the typesetter which font (type style) to use," the publisher's man said. "Just use a '//F' followed by the font number. Every heading, boldface, or command needs the font command."

"That doesn't sound too bad," I ventured.

"Most characters won't give any problems," the publisher's man went on. Of course, there are a few characters the typesetter won't recognize, or that will require some kind of special action."

"Sounds reasonable enough," I said, thinking of German umlauts and French accent graves.

"For example, a left bracket and right bracket can't be used — you'll have to substitute //CHLB and //CHRB instead. The greater than sign and less than sign become //CH23 and //CH34."

"Clever mnemonics on the greater than and less than signs," I joked, noticing for the first time how hot it was in the office.

"And the pound sign, asterisk, back slash, and arrows all need special coding," he went on.

I was starting to perspire. Well, a simple search and replace operation on *My Word*! would do the substitutions with no problem.

"Anything else?" I queried.

"The plus and equals signs typeset just fine," he stated.

"Great!" I said, thanking the deity of my choice and waiting for the punch line.

"Of course, they *do* reset the font in use to F1 — you'll have to add a //F4 or //F5 after each plus or equals, unless you're in bold, in which case you'll have to use //F6."

Now I was *really* sweating, envisioning more time at the keyboard to put in the blasted typesetting commands than to write the material to begin with.

"And, of course, *leading* double quotes must be //", but trailing double quotes must be ".

I wrung out the front part of my shirt, diluting my decaffinated.

"Here's a sample of the manuscript

before inserting the typesetting commands and after," the publisher's man went on.

I glanced down at the proffered samples, in anticipation, and then in horror, as I compared them. What I saw is shown below:

(The BASIC Word Processing Text)
LET Statement
SYNTAX:
LET variable=expression
APPLICATION:
Used to assign a value to a numeric variable or equate a string variable to a string constant or variable.

EXAMPLES:
100 LET ARC=12.34*NEWT
Numeric variable A is set equal to 12.34 times the value of variable NEWT.
200 LET NEXTN$=NAME$+"/"
String variable NEXTN$ is set equal to the string formed by NAME$ plus "/".

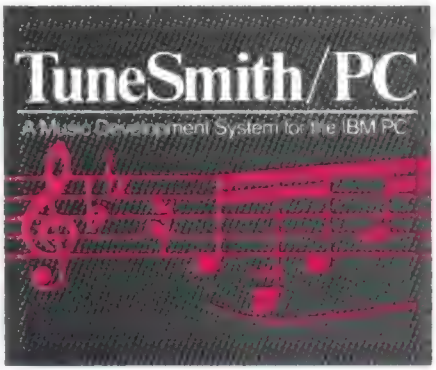
(The Text After Typesetting Commands Were Added)

```
//LT
//UF1LET Statement//LT
//UFR//LT
^^^^LET variable=//F3expression//LT
//UF2//LT
//UF3APPLICATION://LT
//UF4Used to assign a value to a numeric variable or equate a string variable to a string constant or variable.
//LT
//LT
//UF3EXAMPLES://LT
//LT
//UF5100 LET ARC=//F312.34//CHPSNEWT//LT
//UF4Numeric variable A is set equal to 12.34 times the value of variable NEWT.//LT
//UF5200 LET NEXTN$=//F3NAME$+//F3//"/"//LT
//UF4String variable NEXTN$ is set equal to the string formed by NAME$ plus //"/".//LT
```

What had I gotten myself into? I wondered, as I stumbled from the office. Should I simply cancel the contracts? This was editing beyond the call of duty . . .

However, later, in the cool, reflective recesses of my computer room, sequential file processing came to mind. The manuscript was simply in the form of a text file on disk, wasn't it? That file could be read in by BASIC statements and each line processed. A simple translation would replace special characters with any string, and I could use the logic of a BASIC program to detect special conditions, such as which font was currently in use. I set out to code the processor. Three days later, I stumbled out of the computer room, bloodly yet unbowed, with a program that automatically converted a plain text file into one containing the required typesetting codes. Profound? No. Something that you can do quite easily — I'll show you how. You can use the same technique for replacing BASIC commands, such as PRINT to LPRINT, for embedding printer control codes for various printers in text files to be printed, for counting words and making

"Faithful renditions...we were astonished!"
—**James Langdell,** *PC Magazine*



Perfect Pitch for PC

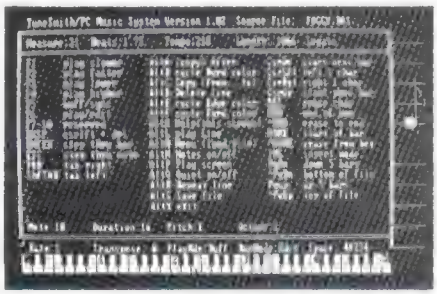
The Premier Music Composition System for the IBM PC

TuneSmith/PC™ is the first true music development software for your IBM PC or compatible. It can create and play music with a melody line played above harmonic chords.

FOR THE MUSIC STUDENT,
TuneSmith/PC™ can help you study music theory. Combine compositions with games and other programs, or make the PC part of your electronic band.

FOR THE COMPUTER HOBBYIST,
TuneSmith/PC™ will turn your PC into a musical instrument. It's a great way for the whole family to discover their hidden musical talents. Ten sample tunes included.

FOR THE ARMCHAIR COMPOSER,
TuneSmith/PC™ plays back musical passages in a variety of tempos and pitches. Forty special function keys make composing faster and easier. Built-in HELP screen:



The system includes a complete spectrum of commands for every music editing function & provides a continuous display of information about the music being played. Friendly beeper warns you of musical errors.

Requires IBM PC, XT, AT, Jr. or compatible with 96K, 1 single-sided drive, 80 column display.

MAIL ORDER OFFER

$49.95

(plus $5.00 postage and handling)

For Visa, MasterCard or American Express call 312-236-8473-ext PCM

Or send check or money order to:
BLACKHAWK DATA CORP.
307 North Michigan Avenue, Dept. PCM
Chicago, Illinois 60601

up lists of words from text files, for changing upper- to lowercase and vice versa, and for any special processing that operates with a text file.

Sequential File Makeup

All BASIC files, almost all word processing files, and many other files are sequential files. Sequential files have to be read from beginning to end — you cannot simply go into the file and pick out the 100th record, as you can with random files. Although at first glance it would seem that random files have a lot to offer because of the ability to pick out any record, sequential files are usually more compact than random files. In addition, many applications lend themselves to a sequence of data — word processing lines and BASIC lines, for example. In these applications and many others, you *want* to read through the file from beginning to end.

Each file is made up of a number of *records*. Imagine a personnel file folder containing a separate form for each employee at a company and you have the idea. The form (record) itself might contain *fields* for "name," "address," "phone number," "social security number" and the like. These fields are very well defined and usually fixed-length in a random file, but may be variable length in a sequential file record. Of course, for some files the records contain no fields, as is the case of a word processing file.

Generally, a record in a sequential file corresponds to a single line on the screen or several screen lines up to 255 characters.

All of the characters in a sequential file are typically ASCII characters. ASCII characters have codes of 0 through 127. Most of the characters in this range are "normal" text characters such as A, z, 7, &, or N. The characters from 0 through 31 are special "control code" characters which are displayable as those funny "happy faces" and similar symbols. These are not standard symbols for the control codes, but just the ones IBM happened to like. Certain control code characters cause display or printer actions, such as 7 (beep), 9 (tab), 10 (line feed), 11 (home), 12 (form feed), 13 (carriage return), 28 (cursor right), 29 (cursor left), 30 (cursor up), and 31 (cursor down).

The codes from 127 through 255 are not ASCII codes, but will display, either as "European" characters, graphics or line characters, Greek symbols, or special symbols. These codes are also allowed in sequential files. Any character in a sequential file can be a value of 0 through 255, all possible combinations of 8-bit data values. Any disk file can be TYPEd out to display its contents, but files containing non-standard codes will display as garbage, although given an infinite number of Tandy 1000s and an infinite number of Tandy product managers, "Canterbury Tales" would probably appear on the screen around July of 1995.

Tricks with Sequential Files

One of the easiest ways to generate a sequential file is to use the COPY command in MS-DOS:

```
A>COPY CON FILE1
```

After entering this command, every keyboard line will generate a disk file record for file "FILE1." Pressing ENTER at the end of each line causes the next record to be written out. Pressing the F6 function key followed by ENTER will terminate the file. The CON is the mnemonic for the system "console" device, in this case the keyboard.

"Redirection" of output devices in MS-DOS will also create sequential ASCII files. To create an ASCII file that contains a directory listing, for example, you'd do:

```
A>DIR >FILE1
```

The resulting file would contain all of the character data that would normally appear on the screen. This data could then be processed as a sequential file. You might, for example, merge and alphabetize several directory listings using the sequential file techniques we're about to discuss.

BASIC programs can be stored as sequential ASCII files by using the ",A" option when saving programs to disk. The resulting files can be edited by a word processing program, *provided that your word processor can read in pure ASCII files and write out pure ASCII files.* Remember that all BASIC lines must be on one logical line when editing BASIC files. This edited BASIC line,

```
3100 PRINT#2,TT$(J);: PRINT "X":
ZA$=RIGHT$(ZA$,LEN(ZA$)-
LEN(ZB$)): IF (ZA$="")
AND (ZD=1) THEN 3200
```

on four separate logical lines, will give a "Direct statement in file" error when you try to load the edited BASIC

program as the second line is encountered.

EDLIN also produces sequential ASCII files that can be further manipulated by BASIC or the proper word processing program.

Reading in Sequential Files

The steps for reading in a sequential file in BASIC are simple. First, the file must be opened by a BASIC OPEN statement. Next, a LINE INPUT# can be done to read in a record of the file (usually a single screen line) at a time. Finally, a CLOSE is done to properly terminate the file operation. Here's a piece of code that will open a given file and display all the records on the screen:

```
100 INPUT "File Name:",FILNM$
110 OPEN FILNM$ FOR INPUT AS #1
120 IF EOF(1) THEN 160
130 LINE INPUT#1,RECORD$
140 PRINT RECORD$
150 GOTO 120
160 CLOSE
```

In fact, this program is really just an emulation of the DOS TYPE command in BASIC. The "#1" refers to a buffer number associated with a file. Several files may be opened at once, each with its own "buffer" or storage area. References for reads or writes are then done to the buffer number and not to the file name.

The LINE INPUT#1 command reads in the next record of the file OPENed for buffer number one. Commas and other characters, which would ordinarily delimit variables in a sequential data file are read in as well. LINE INPUT is a kind of "read everything on the line" command. The EOF command is a special command that senses the last record of the file. If the last record has already been read, the EOF function is true, otherwise it's false. EOF is an automatic way of detecting the end of a file without having to know the actual number of records in the file, or some special code associated with the end of the file.

The code above is only equivalent to TYPE, but here's the difference — any amount of processing can be inserted after the LINE INPUT. You could, for example, avoid printing lines starting with the characters "REMARK:"

```
100 INPUT "File Name:",FILNM$
110 OPEN FILNM$ FOR INPUT AS #1
120 IF EOF(1) THEN 160
130 LINE INPUT#1,RECORD$
140 IF LEFT$(RECORD$,6)="RE-
MARK" THEN 150 ELSE PRINT
RECORD$
150 GOTO 120
160 CLOSE
```

Writing Out Sequential Files

Writing out sequential files is about as easy. Here's a code segment to write out each keyboard line as a disk file:

```
100 INPUT "File Name:",FILNM$
110 OPEN FILNM$ FOR OUTPUT AS
#1
120 LINE INPUT; RECORD$: PRINT
130 IF RECORD$="***" THEN 150
ELSE PRINT#1,RECORD$
140 GOTO 120
150 CLOSE
```

Again, the file is OPENed, but this time for output. A PRINT#1 then writes the entire line entered from the keyboard as the next record. The CLOSE here is absolutely necessary, as it "flushes" the buffer of the last remnants of data. (Records are "blocked" for disk read and write operations.)

Generally, any BASIC statement that will screen display or read operations from the keyboard can be turned into an equivalent disk file operation simply by attaching a pound sign to the statement and adding a buffer number. The character data will be sent to or from the disk instead of the screen or keyboard.

Simultaneous Reads and Writes

The key to processing sequential files is to do both a read and write operation. This can be done by using two buffers, one for the read and one for the write. An example is shown below. This code opens one file, reads in records one at a time, and writes out records to a second file, throwing "CAUTION" records to the winds. After the last file has been read, the code closes both files by a blanket CLOSE:

```
100 INPUT "Input File Name:",F1$
110 INPUT "Output File Name:
",F2$
120 OPEN F1$ FOR INPUT AS #1
130 OPEN F2$ FOR OUTPUT AS #2
140 IF EOF(1) THEN 180
150 LINE INPUT#1,RECORD$
160 IF LEFT$(RECORD$,7)="CAU
TION" THEN 150 ELSE PRINT#2,
RECORD$
170 GOTO 140
180 CLOSE
```

The resulting second file will be identical to the first except that all

Submitting Material to PCM

Contributions to PCM are welcome from everyone. We like to run a variety of programs which will be useful/helpful/fun for other Tandy Portable and MS-DOS computer owners. We now support the Model 100, the Tandy 200, and the Tandy models 2000, 1200 and 1000.

Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs. All programs should be supported by some editorial commentary explaining how the program works. Generally, we're much more interested in how your submission works and runs than how you developed it. Programs should be learning experiences.

Pay for submissions is based on a number of criteria. The rate of remuneration will be established and agreed upon prior to publication.

For the benefit of those who wish more detailed information on making submissions, please send a SASE to: Submissions Editor, PCM, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit programs or articles currently submitted to another publication.

If you feel qualified to review software and/or hardware products for computers covered in PCM, send us your name, address and phone number; we will send you a questionnaire form and a copy of our reviewer guidelines.

records starting with the letters "CAUTION" will have been deleted from the second file.

What you do between the LINE INPUT# and PRINT# commands is up to you. You may process the records any way you wish, scanning for and reporting on character strings, substituting characters, deleting characters, and the like. Remember that most files will have records of variable length. Use the LEN statement to find the input record length, and then process the records using the LEFT$, RIGHT$, and MID$ string commands.

A Translate Program

I'd repeat the program for converting manuscript text into typesetting here, but it's much too long and not generic enough, unless you happen to have a Itek 1200 typsetter. However, I do have a modest example of another program that uses this concept and might prove valuable to you. It's called *TRANSLTE*, and is shown in Listing 1.

TRANSLTE translates characters from an input disk file to characters for an output disk file. Using *TRANSLTE*, you can do a "search and replace" on all occurrences of "dross," changing it to "gold." You can also do a delete of any character string, deleting all occurences of "windy weather," for example. In addition, you can do the search and replace or search and delete for up to 255 character strings. Each character string defined in the search or replace string may be from one to 32 characters long. One example of how *TRANSLTE* might be useful is in reformatting text files for printing. All characters that are underlined could be changed to bold printing easily with *TRANSLTE*. In addition, all 255 character sets can be redefined to another set of 255 characters by making the search and replace strings only one character long, enabling a simple translation or encryption of a file.

I'll first describe the operation of *TRANSLTE* and then describe some of the thornier points for your own implementation of this type of program.

TRANSLTE uses two tables established by the user. One table is called the "From" table and the other is called the "To" table. Each of the two tables is 255 entries long or smaller. Each entry of each table is 0 to 32 characters (bytes) long. The From table defines characters to be found and the To table defines characters to be used to replace the From characters. A null entry in the To table deletes the corresponding From string.

After *TRANSLTE* is loaded, a menu of six items appears. The first item establishes the From and To tables. A string of characters can be used for the From characters or a single character can be entered in either decimal (+XX) or hexadecimal (&HXXX) format. The decimal and hexadecimal formats allow characters that cannot be entered from the keyboard. Use a –1 to terminate the From table. From one to 256 entries

The Listing:

```
100 ' TRANSLATION TABLE COPYRIGHT 1985 WILLIAM BARDEN JR.
110 OPTION BASE 0
120 CLEAR: DIM FT$(255): DIM TT$(255): FT$(0)=CHR$(0): TT$(0)=CHR$(0)
130 PRINT "TRANSLATION TABLE"
150 PRINT: PRINT "MENU"
160 PRINT "1. ESTABLISH NEW TABLES"
170 PRINT "2. READ IN OLD TABLES"
180 PRINT "3. TRANSLATE A FILE"
190 PRINT "4. EDIT A FROM OR TO TABLE"
195 PRINT "5. LIST THE FROM AND TO TABLES"
196 PRINT "6. WRITE OUT TABLES"
200 INPUT A: IF A<0 OR A>6 THEN 150
210 ON A GOTO 1000,2000,3000,4000,5000,1200
1000 ' ESTABLISH NEW TABLES
1010 FT=0: TT=0
1020 PRINT "ESTABLISH 'FROM' TABLE? Y OR N": INPUT A$
1030 IF A$="y" THEN A$="Y" ELSE IF A$="n" THEN A$="N"
1040 IF (A$<>"N") AND (A$<>"Y") THEN 1020
1050 IF A$="N" THEN GOTO 1110
1060 FT=1: FT$(0)=CHR$(0)
1070 FOR I=1 TO 255: FT$(I)="": NEXT I
1080 GOSUB 12060
1090 FOR I=1 TO 255: GOSUB 10000: IF ZA$="" THEN 1110
1100 FT$(I)=ZA$: FT$(0)=STR$(VAL(FT$(0))+1): NEXT I
1110 PRINT: PRINT "ESTABLISH 'TO' TABLE? Y OR N": INPUT A$
1120 IF A$="y" THEN A$="Y" ELSE IF A$="n" THEN A$="N"
1130 IF (A$<>"N") AND (A$<>"Y") THEN 1110
1140 IF A$="N" THEN GOTO 150
1150 TT=1: TT$(0)=CHR$(0)
1160 FOR I=1 TO 255: TT$(I)="": NEXT I
1170 GOSUB 12060
1180 FOR I=1 TO VAL(FT$(0)): GOSUB 10000: TT$(I)=ZA$: TT$(0)=STR$(VAL(TT$(0))+1)
: NEXT I
1190 GOTO 150
```

can be put into the table. The To table works similarly.

Menu item six allows you to save the From and To tables as disk files. The extension .TO and .FRM will be used for the files created. Menu item two allows you to read in a previously saved table file. You can have as many translation tables as you like, calling them by different names.

Once the tables have been read in, they can be listed by menu item five. The listing is done side by side so that you can see the From and To strings and check them for the proper strings or values in corresponding locations.

Menu item four allows you to edit a table in memory. Specify an entry number from one to 250 and then enter a string or value as in establishing the table. The editing operates on only one entry at a time.

The heart of the program is menu item three, which translates the From strings or values to the To strings or values. The program asks for both file names and then reads in each record of the From file, one at a time. *TRANSLTE* then searches the record for each From string, If found, the From string is replaced by the corresponding To string, if there is one, or deleted, if no To string exists. As each character in the file is processed, a period is printed if no From value was found, or an 'X' is printed for a replacement.

TRANSLTE should be compiled for faster speed. Processing a 1000-word file with a From table of 10 entries of 15 bytes and a To table of 10 entries of 15 bytes takes about four minutes to complete for the compiled version, but the interpretive version is four times slower.

The Translation Code

The actual translation is done in the 3000 area of *TRANSLTE*. The From and To files' names are first input and an OPEN is done for each. The OPEN here is an alternate form of OPEN - OPEN "O" is equivalent to OPEN XXX FOR OUTPUT and OPEN "I" is equivalent to OPEN XXX FOR INPUT. The second and third parameters are the file number and filename, respectively. File number 1 (buffer 1) is used for the From file and file number 2 (buffer 2) is used for the To file.

Variable ZA$ is then initialized with 32 bytes (characters) from the From file. The INPUT$ statement used here is similar to the LINE INPUT# statement except that it will read *every* character in the file, including control characters such as line feeds (10) and carriage returns (13). This is done to allow any character to be translated. Variable ZA$ holds the current set of From characters to be compared — the number of characters in ZA$ is always 32 for comparisons, as the maximum size of the replace string is 32 characters.

Variable ZB$ holds the current comparison string from the From array. The From array is called FT$. Variable J is used in a FOR loop to scan all of

```
1200   WRITE OUT TO AND FROM TABLES
1210 IF FT=0 THEN 1290
1220 PRINT: INPUT "FROM TABLE NAME";A$
1230 IF A$="" THEN 1220
1240 IF LEN(A$)>8 THEN 1220
1250 IF RIGHT$(A$,4)<>".FRM" THEN A$=A$+".FRM"
1260 OPEN"O",1,A$
1270 FOR I=0 TO 255: PRINT#1,FT$(I):PRINT ".";: NEXT I
1280 CLOSE 1
1290 IF TT=0 THEN 1370
1300 PRINT: INPUT "TO TABLE NAME"; A$
1310 IF A$="" THEN 1300
1320 IF LEN(A$)>8 THEN 1300
1330 IF RIGHT$(A$,3)<>".TO" THEN A$=A$+".TO"
1340 OPEN"O",1,A$
1350 FOR I=0 TO 255: PRINT#1,TT$(I): PRINT ".";: NEXT I
1360 CLOSE 1
1370 GOTO 150
2000 ' READ IN OLD TABLES
2010 PRINT: INPUT "READ IN 'FROM' TABLE. Y OR N";A$
2020 IF A$="y" THEN A$="Y" ELSE IF A$="n" THEN A$="N"
2030 IF (A$<>"Y") AND (A$<>"N") THEN 2010
2040 IF A$="N" THEN 2130
2050 ON ERROR GOTO 12010
2060 INPUT "FROM TABLE FILE NAME";A$
2070 IF RIGHT$(A$,4)<>".FRM" THEN A$=A$+".FRM"
2080 IF LEN(A$)>12 THEN 2060
2090 OPEN"I",1,A$
2100 FOR I=0 TO 255: INPUT#1,FT$(I): PRINT ".";  : NEXT I
2110 CLOSE 1
2120 FT=1
2130 PRINT: INPUT "READ IN 'TO' TABLE. Y OR N";A$
2140 IF A$="n" THEN A$="N" ELSE IF A$="y" THEN A$="Y"
2150 IF (A$<>"Y") AND (A$<>"N") THEN 2130
```

the From strings to find a possible match for the 32 characters in ZA$. Variable ZB$ is loaded with each of the From strings in turn for comparison with the string in ZA$. The comparison is made based upon the length of the From string in variable ZB$ (IF ZB$<>LEFT$(ZA$,LEN(ZB$))).

If the strings match, a From string has been found in ZA$. In this case, the corresponding To string in the To array, TT$ is PRINTed out (to buffer 2), and an 'X' is displayed on the screen. The From characters in ZA$ are then discarded, starting from the left. This leaves from 31 to 0 characters in ZA$. ZA$ must always have 32 characters for comparisons, so a series of INPUT$(1,#1) of one character each are done to fill out ZA$ to 32 characters once more. The search then restarts again, from the "top" of the From table for the new set of characters in ZA$.

If no strings match the characters in ZA$ after a scan of the From array, the leftmost character of ZA$ is PRINTed (PRINT#2,LEFT$(ZA$,1)), a period is displayed for the no match condition, and the leftmost character from ZA$ is deleted. One character is then read in and appended to the right hand side of ZA$ by an INPUT$ statement. ZA$ is now 32 characters long once more, ready for the comparsion.

Variable ZD is used to flag an EOF, or end-of-file condition.

This process continues for the entire From file, with one character at a time rounding out the 32-character block for comparisons, and the entire From array being scanned for each new set of characters. There is a great deal of "overhead" in the logic, but the process executes fairly rapidly regardless. A conversion to assembly language, however, would speed up the execution by 50 times or more!

Similar processing can be done for your own applications, using the basic input/output structure described earlier in the article.

Next month we'll have more to say about MS-DOS applications on the Tandy 1000, 1200, and 2000.

```
2160 IF A$="N" THEN 2240
2170 INPUT "TO TABLE FILE NAME";A$
2180 IF RIGHT$(A$,3)<>".TO" THEN A$=A$+".TO"
2190 IF LEN(A$)>11 THEN 2170
2200 OPEN"I",1,A$
2210 FOR I=0 TO 255: INPUT#1,TT$(I): PRINT ".";: NEXT I
2220 CLOSE 1
2230 TT=1
2240 ON ERROR GOTO 0
2250 GOTO 150
3000 ' TRANSLATE FILES
3010 ZD=0
3020 INPUT "NAME OF FILE TO BE TRANSLATED"; A$
3030 ON ERROR GOTO 12010
3040 OPEN"I",1,A$
3050 ON ERROR GOTO 0
3060 INPUT "NAME OF FILE TO RECEIVE TRANSLATION";B$
3070 IF A$=B$ THEN PRINT "CANNOT BE SAME NAME": GOTO 3060
3080 OPEN"O",2,B$
```

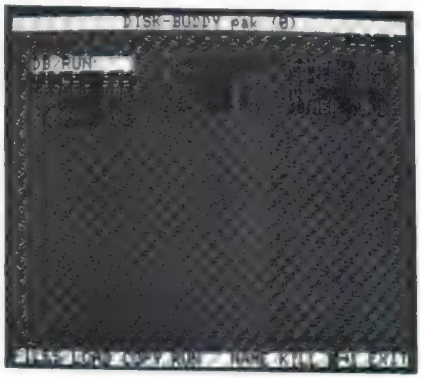

Civilize the DISK/VIDEO INTERFACE • MAKE SAVAGE THE MODEL 100

With DISK-BUDDY software for your DV/I, only function keys need touch your disk files!

Twelve programs for one low price, work together to give the TRS-80 Model 100 unique disk file handling abilities. Access disk files from menu, just like a RAM file. Commands you were tediously typing become simple function key selections. And that's just part of the power of the DISK-BUDDY PAK. It does things you can't:

- Create "superfiles" the length of the disk, over 170,000 bytes!
- Print out, or display on either screen, or copy to disk or other device, a disk data file *of any length*, **WITHOUT USING ANY ADDITIONAL RAM!**
- Furthermore, transfer records can be "filtered", an example:

  An address file on disk can contain more than 1300 entries. You can move to RAM, or screen, or printer, etc., those records which contain "California" and "Tennis" or whatever. Four logical modes of search are available.



- Also: Features to run, load, save, sort, kill, measure, append, and name, disk files and RAM files, and much more.

PCM Magazine (March, '85) said: *". . . an excellent buy" . . . "manual is extremely well written" . . . "will substantially increase the speed and ease of operation of the M-100 with the D/VI."*

That was the idea in the first place.

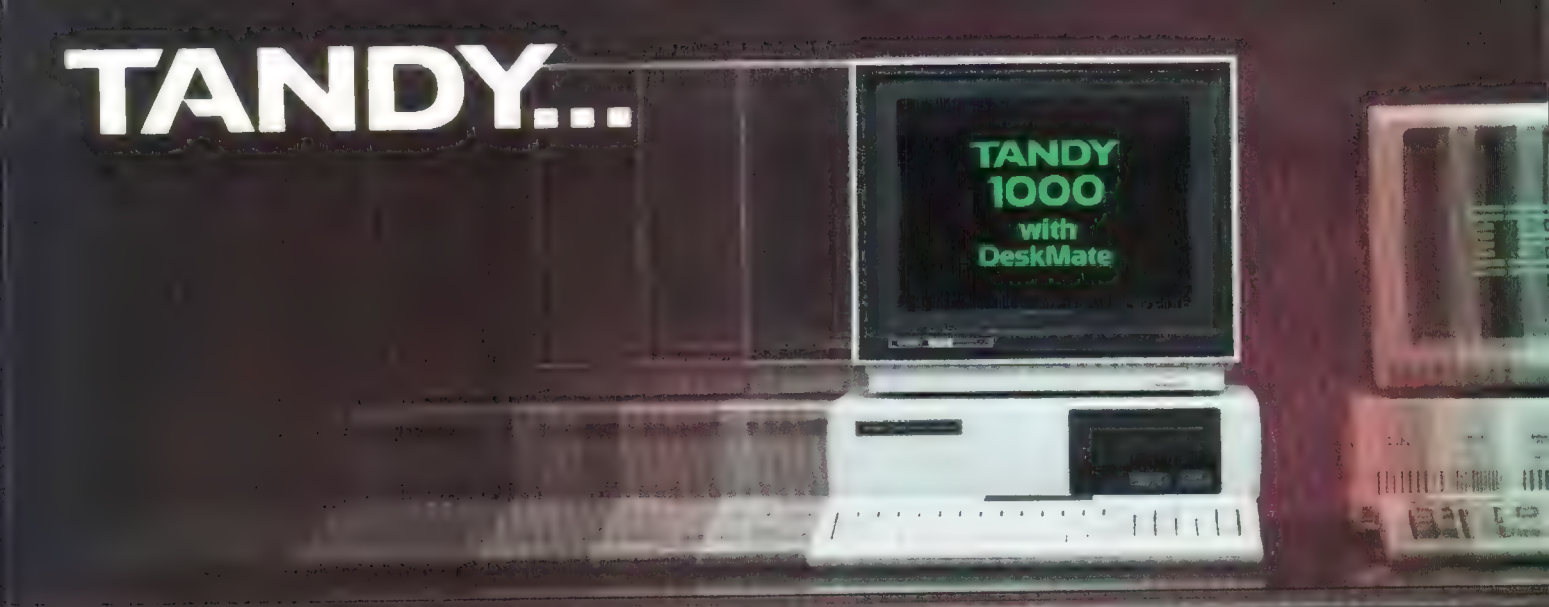
TO ORDER THE DISK-BUDDY PAK (disk and manual): Send name, address, and check or money order for $39.95 plus $2.00 postage and handling to the address below. (For COD's call: 212-243-2129)

BuddySystems
220 West 24th Street
New York, NY 10011

```
3090 ZA$=INPUT$(32,#1)
3100 FOR J=1 TO VAL(FT$(0)):ZB$=FT$(J): IF ZB$<>LEFT$(ZA$,LEN(ZB$)) THEN 3150
3110 PRINT#2,TT$(J);: PRINT "X";: ZA$=RIGHT$(ZA$,LEN(ZA$)-LEN(ZB$)): IF (ZA$="")
 AND (ZD=1) THEN 3200
3120 PRINT "X";: IF ZD=1 THEN 3100
3130 FOR K=1 TO 32-LEN(ZA$): IF (EOF(1)) AND (ZA$="") THEN 3200 ELSE IF EOF(1) T
HEN ZD=1: GOTO 3100
3140 ZA$=ZA$+INPUT$(1,#1): NEXT K: GOTO 3100
3150 NEXT J
3160 PRINT#2,LEFT$(ZA$,1);: PRINT ".";: ZA$=RIGHT$(ZA$,LEN(ZA$)-1)
3170 IF ZA$="" THEN 3200 ELSE IF ZD=1 THEN 3100
3180 IF EOF(1) THEN ZD=1: GOTO 3100
3190 ZA$=ZA$+INPUT$(1,#1): GOTO 3100
3200 PRINT: PRINT "FILE TRANSLATED"
3210 CLOSE 1,2
3220 GOTO 150
4000 ' EDIT ONE OR BOTH TABLES
4010 PRINT: INPUT "FROM TABLE (FT) OR TO TABLE (TT)";A$
4020 IF A$="ft" THEN A$="FT" ELSE IF A$="tt" THEN A$="TT"
4030 IF A$<>"FT" AND A$<>"TT" THEN 4010
4040 GOSUB 12060
4050 PRINT: INPUT "ENTRY # 1 TO 255";I
4060 IF I=-1 THEN GOTO 150
4070 IF I<0 OR I>255 THEN 4050
4080 GOSUB 10000
4090 IF A$="FT" THEN FT$(I)=ZA$: IF VAL(FT$(0))<I THEN FT$(0)=STR$(I)
4100 IF A$="TT" THEN TT$(I)=ZA$: IF VAL(TT$(0))<I THEN TT$(0)=STR$(I)
4110 GOTO 4050
5000 ' LIST THE FROM AND TO TABLES
5010 PRINT "ENTRY";TAB(7);"FROM";TAB(40);"TO"
5020 FOR I=1 TO VAL(FT$(0)): PRINT TAB(3);I;TAB(7);FT$(I);TAB(40);TT$(I): NEXT I
5030 GOTO 150
10000 ' INPUT VALUES SUBROUTINE
10010 PRINT "ENTRY #";I
10020 ZA$=""
10030 FOR ZJ=1 TO 32
10040 INPUT ZB$
10050 IF ZB$="-1" THEN 10120
10060 IF LEFT$(ZB$,2)="&H" THEN GOTO 10080
10070 IF ZJ=1 AND LEFT$(ZB$,1)<>"+" THEN ZA$=LEFT$(ZB$,32): GOTO 10120
10080 ZA=VAL(ZB$)
10090 IF (ZA<0) OR (ZA>255) THEN PRINT "ERROR. REENTER FROM START": GOTO 10010
10100 ZA$=ZA$+CHR$(ZA)
10110 NEXT ZJ
10120 RETURN
12000 ' FILE NOT FOUND ERROR PROCESSING
12010 IF ERR<>53 THEN ON ERROR GOTO 0: RESUME
12020 PRINT "FILE NOT FOUND"
12030 IF ERL=2090 THEN RESUME 2060
12040 IF ERL=2200 THEN RESUME 2170
12050 IF ERL=3040 THEN RESUME 3020
12060 ' PROMPT FOR INPUT VALUE
12070 PRINT "INPUT VALUES, EITHER IN NUMERIC FORM OR AS A CHARACTER"
12080 PRINT "STRING. IF IN NUMERIC FORM, PREFIX A +SIGN. A -1 ENDS"
12090 PRINT "ENTRY. IN EITHER FORM A -1 ENDS TABLE. UP TO 32 CHAR-"
12100 PRINT "ACTERS OR VALUES MAY BE USED."
12110 RETURN
```

PCM

TANDY...



Clearly

Superior™

We edged out the competition with better solutions...across the board.

We work harder for your business.

At Radio Shack Computer Centers, you'll find more than just computers. You'll find the *right* computer, backed with a complete line of software and accessories, as well as the best in customer support.

The Tandy 1000: more features than an IBM PC, for only $999.

This is the one you've heard about. It's compatible with the IBM PC, for one of the widest selections of software around.

Your first six programs are on us.

Unlike the PC, the Tandy 1000 comes with its own software—DeskMate.

DeskMate combines word processing, electronic filing, spreadsheet analysis, scheduling, telecommunications, and electronic mail on one disk. So you can get right to work the very first day.

Plus, the adapters you'll need to use a monitor, printer, joysticks and light pen are all built in. These adapters, as well as MS-DOS and BASIC that also come with the 1000, are all extra cost options on the IBM PC.

Or start with ten-megabytes for only $1999.

The Tandy 1000 is also available with 256K and a built-in hard disk system for fast access to ten million characters of information.

Best of all, the Tandy 1000 HD computer costs the same as an IBM PC *without* a hard disk!

The Tandy 1200 HD: a "mirror image" of IBM's PC XT for $1999.

If you're using the IBM PC XT, its $3895 price tag is probably doing a good job of holding back your expansion plans. What can you do about it?

Expand affordably.

The Tandy 1200 HD uses virtually every software package made for the PC XT. It's expandable with the same hardware. Even the keyboard follows the same layout, with improved placement of the shift keys.

So what's the difference?

It's simple. The Tandy 1200 HD costs $1,896 *less* than the PC XT. And we include the printer adapter, an option on the XT.

The Tandy 2000: the performance of IBM's PC AT for $1599.

Like the IBM PC AT, the Tandy 2000 offers two to three times the speed and two times the disk storage of IBM's PC.

The Tandy 2000 also offers twice the graphics resolution (640 × 400), with twice the colors of the IBM PC, for color charts and graphs that are nothing short of spectacular.

The bottom line.

At $3999, a PC AT is $2400 more than a 2-disk, 256K Tandy 2000. In fact, you can get a complete Tandy 2000 system with color monitor and graphics for just $2757.90—still over $1000 less than an AT *without* a monitor!

The closer you look, the better we look.

Now if you're thinking IBM's personal computers cost more because of the support IBM is so famous for, think again.

We offer a free customer support "hotline". IBM's telephone support will cost you $40. *Per call.*

We also offer low-cost delivery and set-up, new-user orientation, as well as on-site training in all major markets. IBM doesn't offer these customer support services to their PC owners.

Get the edge on *your* competition.

For the best in computers, software, training, support service and leasing plans, stop by any Radio Shack Computer Center.

We have what it takes to sit down and talk business.

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Send me a free copy of your 1985 computer catalog.

Mail to: Radio Shack, Dept. 86-A-152
300 One Tandy Center, Fort Worth, Texas 76102

Name _____

Company _____

Address _____

City _____

State _____ Zip _____

Telephone _____

*Plus applicable use/sales tax. Radio Shack product pricing applies at Radio Shack Computer Centers and at participating stores and dealers. MS is a trademark of Microsoft Corp. IBM is a registered trademark of International Business Machines Corp. Pricing and service availability as of 5/1/85.

The Gallery

By Wayne Sanders

This month's Gallery program was originally written for use at the upcoming PCMfest. We thought we'd pass it along to you for your enjoyment.

A few interesting techniques are used in the PCM logo program. Because it was made to to run on both the Tandy 1000 and 2000 computers, the differences in the graphics resolutions of the two machines had to be overcome. The solution was to multiply all 'X' and 'Y' values and circle diameters by a variable, SR. If the program is running on a Tandy 2000, SR has a value of one and the 'X,' 'Y' and diameter values are not changed. If, however, it is running on a Tandy 1000, the values are all multiplied by .5 to scale down the graphics for the Tandy 1000's lower resolution.

The SR variable is also used in the timing loop in Line 1420. Since the Tandy 2000 is much faster than the 1000, the variable is used in the loop to adjust the timing. This makes the time delay on both machines roughly equal.

While the Tandy 1000 has lower resolution than the 2000, it is capable of displaying more colors simultaneously. This is taken advantage of by using the CR variable to adjust the number of available colors. CR is seven for the Tandy 2000 and 15 for the 1000.

If you have a Tandy 2000, change the first line of the program to read: `1000 MODEL=2000`.

The Gallery is always seeking new talent. If you have a piece of computer art that you're dying to share with the world, send it along to PCM. You may just see your masterpiece in these pages and a check for $50 in your mailbox!

PCM

THE PERSONAL COMPUTING MAGAZINE
FOR TANDY COMPUTER USERS

The Listing:

```
1000 MODEL=1000
1010 IF MODEL=1000 THEN CLEAR ,,,32768!:SR=.5:SC=5:CR=15
1020 IF MODEL=2000 THEN SR=1:SC=3:CR=7:LOCATE ,,0
1030 CLS:SCREEN SC:KEY ON:KEY OFF:CLS:TS=1.745329E-02:RANDOMIZE(TIMER)
1040 FOR Y=120*SR TO 230*SR STEP 20*SR
1050    LINE (0,Y)-(639*SR,Y),INT(RND*CR)+1
1060 NEXT Y
1070 CC=INT(RND*CR)+1
1080 LINE (120*SR,100*SR)-(155*SR,230*SR),CC,BF
1090 CIRCLE (180*SR,140*SR),50*SR,CC,270*TS,90*TS,.8
1100 CIRCLE (165*SR,140*SR),18*SR,CC,270*TS,90*TS
1110 LINE (110*SR,100*SR)-(180*SR,124*SR),CC,BF
1120 LINE (155*SR,156*SR)-(180*SR,180*SR),CC,BF
1130 PAINT (200*SR,140*SR),CC,CC
1140 LINE (110*SR,210*SR)-(165*SR,230*SR),CC,BF
1150 CIRCLE (300*SR,165*SR),65*SR,CC,90*TS,270*TS,1
1160 CIRCLE (315*SR,165*SR),40*SR,CC,22*TS,334*TS,1.5
```

```
1170 CIRCLE (300*SR,191*SR),65*SR,CC,270*TS,350*TS,.6
1180 CIRCLE (300*SR,139*SR),65*SR,CC,0,90*TS,.6
1190 LINE (340*SR,100*SR)-(365*SR,150*SR),CC,BF
1200 LINE (340*SR,180*SR)-(364*SR,200*SR),CC,BF
1210 PAINT (276*SR,165*SR),CC,CC
1220 LINE (380*SR,100*SR)-(400*SR,230*SR),CC,BF
1230 LINE (370*SR,210*SR)-(410*SR,230*SR),CC,BF
1240 LINE (370*SR,100*SR)-(430*SR,120*SR),CC,BF
1250 LINE (400*SR,150*SR)-(430*SR,230*SR),CC
1260 LINE (430*SR,100*SR)-(450*SR,150*SR),CC
1270 LINE (450*SR,150*SR)-(470*SR,100*SR),CC
1280 LINE (430*SR,230*SR)-(450*SR,230*SR),CC
1290 LINE (450*SR,230*SR)-(502*SR,100*SR),CC
1300 LINE (470*SR,100*SR)-(530*SR,120*SR),CC,BF
1310 LINE (480*SR,100*SR)-(520*SR,230*SR),CC,BF
1320 LINE (470*SR,210*SR)-(530*SR,230*SR),CC,BF
1330 PAINT (450*SR,165*SR),CC,CC
1340 CIRCLE (370*SR,200*SR),10*SR,CC,270*TS,0*TS
1350 PAINT (378*SR,208*SR),CC,CC
1360 CIRCLE (410*SR,200*SR),10*SR,CC,180*TS,270*TS
1370 PAINT (402*SR,208*SR),CC,CC
1380 X$="THE PERSONAL COMPUTING MAGAZINE"
1390 LOCATE 18,40*SR-.5*LEN(X$):PRINT X$
1400 X$="FOR TANDY COMPUTER USERS"
1410 LOCATE 20,40*SR-.5*LEN(X$):PRINT X$
1420 FOR I=1 TO 2000*(SR^2):NEXT I:GOTO 1070
```

PCM

EXPAND THE TANDY 2000®
TO
896K
OF DOS ADDRESSABLE RAM

Includes Ramdisk Software at No Extra Cost ...

Add 640K of External RAM Memory to TANDY 2000's® 256K of Internal RAM Memory for a total of 896K of DOS Addressable RAM.

* Uses only one (1) External Memory Card Slot
* RAMDISK Software is included with the price.

cost: $650.00 for a 640K External RAM Memory Board (Includes RAMDISK Software) *Call about our Trade-in Exchange option if you already use Tandy's 128K External Memory Board.*

for ordering and information call: (509) 627-5291 Hours 9:00-5:00 pst

terms: We accept Check, MO, Visa, Mastercard & COD. Add 7.3% sales tax in Wash. Add $3-shipping/handling, $5 foreign.

TANDY 2000® IS A TRADEMARK OF TANDY CORPORATION

ENVISION DESIGNS
1909 ORCHARD WAY • RICHLAND, WASHINGTON 99352

CDA
COMPUTER SALES
MAIL ORDER WORLDWIDE
SINCE 1977

MAJOR CREDIT CARDS ACCEPTED!

EPSON PRINTER PKG.

•**EPSON** FX-85 DOT MATRIX PRINTER
•ADJUSTABLE FORMS TRACTOR
•TABLE TOP PRINTER STAND
•EXTRA PRINTER RIBBON
•CASE OF PAPER (1800 SHEETS)
•PRINTER DUST COVER
•INTERFACE CABLE
(Specify Computer-Color Computer add $50)

NEW!

The all new EPSON FX-85 dot matrix printer prints at a quick 160 characters per second, bidirectional logic seeking, PLUS a near letter quality mode that prints high quality typewriter style letters. One year warranty!
IDEAL FOR • TANDY 1000 • TANDY 1200HD • MODEL 4

LIST - $599

CDA - $469

ORDER TOLL FREE 800-526-5313

ORDERING INFORMATION AND TERMS:
Mail to: CDA COMPUTER SALES 31 Marshall Hill Rd., West Milford, NJ 07480 Include address and phone number. Shipping, handling and insurance are additional. Add 2% UPS Ground ($3.50 minimum) UPS Blue 6% ($5.00 minimum) **NO C.O.D.** Cashiers checks, moneyorders and credit cards ship immediately. Personal and company checks allow 15 days.
ORDER 9am-7pm EST Monday-Friday, Saturday 10-4
OUR REFERENCES: We have been selling computers since 1977 and in computer Mail Order since 1980. Our bank is: First Fidelity, West Milford, N.J. D&B listed.
ORDERS 800-526-5313 INQUIRIES & IN NJ 201-728-8080
SATISFACTION GUARANTEED!
If you're not 100% satisfied, return within (7) seven days for a full refund

Changes your Model 100 into a totally different computer with capability you never thought possible.

Infoworld rated Lucid's performance "excellent"

PCSG says "Satisfaction Guaranteed or your money back within 30 days!"

LUCID© is here now. It is on a ROM cartridge that snaps into the compartment on the back of your Model 100. It takes no memory to load and no memory for operating overhead. That means you have the full 29.6k bytes free to store your data.

First, LUCID© is memory conserving. It will let you build a large spreadsheet–255 row by 126 column capacity. You build huge spreadsheets in your Model 100's RAM that could consume 80 to 100K on a desktop computer.

Secondly, LUCID© is fast. LUCID© is so rapid, a 36 column corporate financial statement took less than 4 seconds to calculate.

Thirdly, LUCID© has features you won't find in most other spreadsheets. For example, when you type a label (text) it will cross column boundaries; in other words when you type a label or title it will appear as you type it irrespective of column or width. LUCID© also allows you to set column widths individually, and of course LUCID© has insert row and insert columns, as well as other standard features. LUCID© even lets your formulas refer to cells in other spreadsheet files.

Further, LUCID© has what no other spreadsheet has: Cut, Copy, and Paste. It uses the same keys as Cut and Paste in TEXT, but here's the difference: it takes all the formulas with it when you paste and they all automatically recalculate with the entire sheet.

And here is what is really amazing. You can copy or cut from one spreadsheet and paste into another spreadsheet or even a TEXT file.

LUCID©supports all BASIC math functions as well as Log, sine, cosine, tangent, exponentiation and other sophisticated math functions.

LUCID© has so many features that you will say "this is what I need in a spreadsheet", such as automatic prompting of an incorrectly typed-in formula showing just where the mistake was made.

LUCID© has expanded "go to" functions that remember and produce a windowing capability.

But perhaps most remarkable is that LUCID© is not only a spreadsheet but a program generator as well. First, LUCID© lets you protect all cells against entry or change, and then unprotect just the cells you want for someone else to use as input fields.

LUCID© will not only process values, but text input as well so that the facts other than numbers can be responded to. LUCID© has the ability for you to refer in a formula to cells containing words. This feature combines with the capacity of doing "if then" statements that work by doing table lookups against even massive X/Y charts of text or numerical information. You can produce a program that responds to inputs with no programming knowledge whatsoever.

You can prepare a report section in your spreadsheet with instructions to your user for printout, and they can produce a personalized printout that responds to their input. All your formulas and tables that did the calculations and provided the facts are invisible to that user. LUCID© is useful for doctors for patient questionnaires, troubleshooting technicians, purchase clerks, people doing job quotes, stores for customer workups, insurance agents and anybody who needs to process specific facts and numbers to produce a report based on those responses.

LUCID© comes with a manual that explains not only the characteristics of LUCID©, but will train you how to use a spreadsheet even if you have never seen one before. You are shown how to do budgets, forecasts, breakeven analysis amortizations and many other types of personal and business reports and calculations.

User friendly is such an over-used term in this industry, but a typical comment has been "I have never seen a spreadsheet that does so much, and yet LUCID© is so much easier and faster to use."

LUCID© is a result of a most exhaustive developmental effort in which PCSG's objective was to develop a spreadsheet that was better than the state-of-the-art. We are so pleased because LUCID© provides for the Model 100 spreadsheet capability you cannot equal on a desktop computer.

LUCID© is, in our opinion and that of those who have examined it, a breakthrough. We sell it on a 30 day trial. If you are not completely satisfied, return it within 30 days for a full refund. Priced at $149.95, on snap-in ROM. MasterCard, Visa or COD.

© PCSG 1984

First with software for the Model 100

PORTABLE COMPUTER SUPPORT GROUP

1-214-351-0564

11035 Harry Hines Blvd., #207
Dallas, Texas 75229

Let a Spreadsheet Dispel the Gloom of Form Creation

By Richard A. White
PCM Contributing Editor

It occurred to me that some of the utility programs in computer magazines programmed in BASIC might be done as well or better in a spreadsheet. This is not to detract in any way from the quality of the programming from the contributors. Rather, it is an opportunity to demonstrate the power of a spreadsheet in a direct comparison.

I had to look no further than Joseph Kohn's *Purchase Order* BASIC program in PCM's sister publication, the May 1985 RAINBOW. I would guess that 20 to 40 hours of programming were involved.

Forms are where spreadsheets shine, so out came my spreadsheet and within two hours my purchase order form was done, complete with ASCII saves of the results to feed the word processor when it came time to write this column. Well, not quite done, since I did find a few bugs as I put the column together and had a few ideas for improvements. The illustrations accompanying this column are the third or fourth try. Still, no major amount of time was involved.

My version of the purchase order is shown as Figure 1. I started with Joseph Kohn's design and modified it to my taste. Having some of the basic design already done helped.

(Richard White has a long background with microcomputers and specializes in BASIC programming. With Don Dollberg, he is the author of the TIMS *database management program for the Color Computer.)*

To construct the form, we start with column widths. These are determined by the central part of the form where the items being ordered are listed. Column A was set to four characters to form a left margin. Column B was also set to four since I do not expect the quantity to be a very large number.

The beauty of a spreadsheet program is that something like a column width is easily set while the form is being constructed. So, if you are a large volume buyer of, say, disk hub reinforcement rings, and your order quantity is each, you may want to leave the column a bit wider.

There is a fair amount of text on the form. If you change column widths after the text is entered, some text and formatting may be messed up. I said may because this depends on how your particular spreadsheet is coded. Make your column width decisions early and try to stick with them.

Also, don't forget your printer. How many characters wide can you print? The purchase order is best fit onto a standard 8½ by 11-inch sheet using normal type so an 80-character per line limit, including margins, is dictated.

The lot size, which might be a number or each, dozen, ream and the like is in Column C. This was set at six characters. Column D, reference number, was set at 12, which may be on the short side.

I set the description, Column E, to 30 characters. Some spreadsheets set a column width limit. In others, once you

Figure 2 in last month's Basic Bytes contained an error. The long formula in Cell B5 should be: `B2/(1-((1+(B3/12))-B1))/(B3/12))`

Figure 1
Purchase Order Spreadsheet Example

```
   [A ][B ][ C  ][    D    ][              E              ][   F   ][   G   ]
 1-
 2-                                RICHARD A. WHITE
 3-                                   44 DOW CT.
 4-                              FAIRFIELD, OH  45Ø14
 5-                                 513-829-851Ø
 6-
 7-                     DATE:  JUNE 2, 1985
 8-
 9-                       TO:  PORTABLE COMPUTER SUPPORT GRP
1Ø-                            11Ø35 HARRY HIMES BLVD.
11-                            NO. 2Ø7
12-                            DALLAS TEXAS   75229
13-
14-                                 PURCHASE ORDER
15-   ***********************************************************************
16-       LOT   REFERENCE                                        UNIT    TOTAL
17-   QTY SIZE  NUMBER                 DESCRIPTION              PRICE    PRICE
18-   --- ----- ----------- ------------------------------ -------- --------
19-    1  EA                LUCID SPREADSHEET ROM             149.95   149.95
2Ø-
21-
22-
23-
24-
25-
26-
27-
28-
29-
3Ø-   --- ----- ----------- ------------------------------ ------------------
31-                                                      SUB TOTAL   149.95
32-                                                PERCENT  DISCOUNT
33-                                                       DISCOUNT     Ø.ØØ
34-                                                      SUB TOTAL   149.95
35-                                     PERCENT TAX/TAX      Ø.ØØ     Ø.ØØ
36-                                                       SHIPPING     Ø.ØØ
37-                                                                 ---------
38-                                                          TOTAL   149.95
39-   ***********************************************************************
4Ø-
41-   PRICES SOURCED FROM-  APRIL 1985 PCM MAGAZINE
42-
43-
44-
45-   CHARGE TO VISA CARD # ØØØØ 111 222 333
46-
47-       EXPIRATION DATE-  1Ø/86
48-
49-              SIGNATURE-  ----------------------------
5Ø-                          RICHARD A. WHITE
51-
52-   SHIPPING INSTRUCTIONS
```

set a column width it applies to all columns. The ability to individually set column widths is of prime importance and I have previously expressed my displeasure with Model 100 *Multiplan* for failing to do this. As soon as I can find someone to buy my copy, I will make use of the purchase order shown above.

Columns F and G were left in the default nine columns wide.

In the heading, the name and address are positioned in Column E to be about centered on the whole form. In many spreadsheets, you can only right- or left-justify text in a cell with the format command. With these you can manually center text by inserting spaces before the characters. *Lotus 1-2-3* lets you format a cell to center text by preceding the text with a caret. Check out your "helps" to understand your options.

Next, the words DATE and TO: are in Column D and are right-justified. The user can then enter the data following these words in Column E. A trick here is to type a space after both DATE and TO: so they don't position exactly at the right side of the column.

In the order section we are going to do considerable cell formatting. The right-justification format mentioned above is but one of a large number available. The first four column headings are routine without special format. UNIT PRICE and TOTAL PRICE are right-justified to align nicely over the columns of figures.

Items being ordered are entered in rows 19 through 29. Some of the columns in this area carry a number of format controls. The quantity, QTY, is to be an integer justified to the left. This lets the LOT SIZE entries be left-justified with one or more spaces between these entries and the quantity. The format command must be invoked for each format control imposed on the cell.

A left-justification format is applied to both columns C and D so that any numbers entered will be left-justified as text is. Some spreadsheets won't let you impose a format onto an empty cell. They generally give no objection; they just don't comply. You can beat this by putting a space into the cell and then formatting it. Since the space is a non-printing character, the cell looks empty. Whatever you may later enter will wipe out the space.

Columns C and D also want to carry integer format. This is no problem since a cell may have a number of formats at any time.

Some spreadsheets will not let you format over a range using the format command. The way around this is quite simple and may be easier than formatting ranges of cells. In our example, put spaces in cells B19, C19 and D19, then apply all the required formats to these cells. Finally, replicate the range B19 . . . D19 over the range B20 . . . B29. The range B19 . . . D19 is then replicated 10 times, each time starting in another 'B' column cell.

The total price for the line entry is the quantity in Column B times the unit price from Column F. The formulae to calculate total price in *VisiCalc/Lotus 1-2-3* format are shown in Figure 2.

An IF statement is used to test if there is a value in Column B before making a calculation. If there is none, the contents of Cell A1 are written. A1 contains a space so it looks and prints like there is nothing in the cells in Column G corresponding to empty purchase order entries. Figure 2 is an expansion of the IF statement to help those with other spreadsheets translate to their system.

I need now to remind you about relative and absolute addressing (in case you did not catch last month's offering and study the subject). In G19, cell references B19 and F19 should be considered as relative addresses. When the formula in G19 is replicated or copied to cells G20 . . . G29, which I fully expect you to do rather than doing a lot of error-prone typing, B19 and F19 should change line numbers. The replicated formula in G20 should read IF(B20>0,B20*F20,A1). Most spreadsheets will adjust line numbers for you.

We are still left with a problem; since A1 needs to remain A1, it is an absolute address. Spreadsheets have different ways of dealing with this problem such as asking you if the formula is to be adjusted to *Lotus 1-2-3's* inscrutable codes that I have not figured out yet. In the worst case, you can always edit lines which is better than retyping, but not much.

Figure 2
***VisiCalc/Lotus 1-2-3* Formulas**
For the Purchase Order

```
 IF(  B19>Ø    ,      B19**F19          ,      A1
    TEST TRUE?  THEN SHOW THIS IN CELL  ELSE SHOW THIS

   [   F   ][          G          ]
15-*******************
16-     UNIT                    TOTAL
17-    PRICE                    PRICE
18- -------- --------
19- 399       @IF(B19>Ø,B19*F19,A1
2Ø-           @IF(B2Ø>Ø,B2Ø*F2Ø,A1
21- 15        @IF(B21>Ø,B21*F21,A1
22-           @IF(B22>Ø,B22*F22,A1
23- 99.95     @IF(B23>Ø,B23*F23,A1
24-           @IF(B24>Ø,B24*F24,A1
25-           @IF(B25>Ø,B25*F25,A1
26-           @IF(B26>Ø,B26*F26,A1
27-           @IF(B27>Ø,B27*F27,A1
28-           @IF(B28>Ø,B28*F28,A1
29-           @IF(B29>Ø,B29*F29,A1
3Ø-------------------
31-SUB TOTAL @SUM(G18...G3Ø)
32- DISCOUNT
33- DISCOUNT G31*(G32/1ØØ)
34-SUB TOTAL G31-G33
35-          G34*(F35/1ØØ)
36- SHIPPING
37-          ---------
38-    TOTAL G34+G35+G36
39-*******************
```

The @SUM(G18 . . . G30) function's range deserves mention. It spans from the upper to the lower dashed lines. This way, any way you are likely to insert rows to enlarge the order entry area will fall within the range and be included. If you add rows, you are likely to remember to replicate the 'G' column formula into those rows, but probably would forget to check the summation.

A likely action is to put the cursor onto Row 30 and do an Insert, which will move Row 30 to Row 31. The *Calc* program will adjust the sum to @SUM(G18 . . . G31) and your new row will be included. The remaining formulae are completely straightforward.

The remainder of the purchase order is also fairly simple. Note that a line saying CHECK ENCLOSED could be included instead of or as well as charge card information. You will probably use one or the other and delete the row or rows that do not apply. The CHECK ENCLOSED was in cells B43 . . . D43 and was blanked out in Figure 1.

Once you have gotten your purchase order shell working, blank out the data only and save it. Thereafter, when you prepare an order, load the empty shell and go to work. PCM

ALWAYS AT SALE PRICES
TANDY COMPUTERS
WITH THE MANUFACTURERS WARRANTY
CHECK OUR SALE PRICES
COMPUTER CENTER
CUTOMER SERVICE
(817) 481-SALE
(9 a.m. - 5 p.m. C.S.T.)
DFW COMPUTER CENTER
326-MAIN ST.,
GRAPEVINE, TEXAS 76051
HARD DISK DRIVES
MODEL 12
HAVE YOU GOT OUR PRICES?
CALL TOLL FREE
1-800-433-SALE
TANDY 1200
TANDY 200
TANDY 6000
MODEL 4-P
TANDY 1000
Radio Shack
CITILINE
VISA
MasterCard

TANDY
ARCNET
FRED W. CRAWFORD
© 1985

Connecting to ViaNet . . . the year of the Network

By Danny Humphress
PCM Technical Editor

As far as earth-shattering computer innovations go, 1985 has been a pretty dull year. Not that we haven't seen a lot of great new equipment introductions, such as the Tandy 1000, but as the microcomputer industry matures, most of the real breakthroughs seem to be past us. Computer innovations today, while remarkable, pale against the leaps in technology we saw in the early years of the microcomputer revolution.

If you had to classify the year 1985 in the microcomputer industry, though, it will probably be remembered as the year of networking. While many companies are offering incompatible networking hardware and software, IBM introduces its complicated grand networking scheme with all the arrogance which we have come to expect from IBM, and Apple is putting its chips on the simple and slow *avant-garde* Macintosh Office network. Meanwhile, back in Fort Worth, Tandy quietly introduces their new PC network, ViaNet.

Unlike most of the others, the Tandy network is compatible with all of Tandy's MS-DOS and Xenix computers as well as the world of IBM PCs and compatibles. ViaNet is simple to connect and use and it does not make all of your software obsolete. It becomes the liaison between IBM, Tandy, and other PCs in your office.

What is Networking?

All of the industry's excitement about networking is not unfounded. Networking allows many computers to simultaneously share resources — resources such as hard disk drives and high speed printers which would be costly or nearly impossible to purchase separately for each individual computer that may need access to them. In addition to the sharing of hardware resources, networking allows different computers to share software and data.

It used to be that the only microcomputer alternative for installations consisting of many users needing access to the same data was to install a single computer with multiple terminals. Not only did the computer have to provide disk storage and printer access to all of the terminals, it had to do the computing as well.

In a local network, on the other hand, each terminal is actually a computer. Therefore, your terminal is not slowed down by other people using the computer. The only thing that is being shared is hard disk drives and printers.

The main advantage to a multi-user computer as opposed to a network is that the multi-user system's terminals are inexpensive "dumb" terminals. A network requires a complete computer for each terminal — although the computer does not necessarily need to have a printer or disk drive connected directly to it. As the costs of microcomputers continues to drop, though, this advantage is quickly disappearing.

(Danny Humphress, PCM's Technical Editor, is the owner of a computer software and consulting firm in Louisville, Ky. Danny brings to PCM his extensive experience with small business computers and applications software.)

Tandy's Networking History

ViaNet is not Tandy's first move into networking. In early 1983, they began delivering a system based on Datapoint's Arcnet for their Model II 8-bit computer.

Model II Arcnet was not well received. Like IBM's PC Network, Model II Arcnet required that at least one computer in the network be a network "server." That is, the one computer would do nothing but route information through the network. It could not be used as a terminal, but must have network server software loaded and running at all times. Because of the slow speed of the 8-bit computers and the software being used, accessing a hard disk across the network was almost like going back to the days when ultra-slow cassette tape was the only available storage medium. Additionally, most existing Model II software had to be rewritten or modified to work in the network — no such software ever saw a store shelf.

Mercifully, Model II Arcnet soon faded into oblivion.

The ViaNet network is based upon the same Datapoint Arcnet hardware standard — one of the fastest and most versatile there is — yet it is the software which makes all the difference.

While ViaNet is often used to describe the entire network system, it is actually just the network software. The hardware is Standard Microsystems Corporation's implementation of the Arcnet standard, and the software comes from ViaNetix, Inc.

With ViaNet software, there is no need for a centralized network server. Any user can access any disk drive (hard or floppy), any printer, any display and any RS-232 device in the network, providing that permission is granted from the "owner" of that device or the system administrator. Hard disk drives and printers can be centralized or scattered about the system — it makes no difference.

How it Works

The Hardware

Each computer in a ViaNet system must have one $499 Arcnet board. There are three versions of the board. One for the Tandy 1000, 1200HD, or other IBM compatible; one for the Tandy 2000; and one for the Tandy 16 or 6000.

Each board is assigned a different "node" number, from 1 to 255, by setting DIP switches on the board. Therefore, the number of computers in any single network is limited to 255.

Since the cabling and hubs for the ViaNet network are the same as Tandy's old Model II Arcnet, Radio Shack Computer Centers have begun to blow the dust off their old Arcnet cables and hubs. They have become salable merchandise again!

The computers are connected together with RG62 coaxial cable. Cables with BNC connectors are available from Tandy in lengths of 10, 20, 50 and 100 feet. Bulk cable and connectors are also available for longer lengths.

If there are just two computers in a network, a single cable between them is all that's needed. One end of the cable plugs into the Arcnet board of one computer and the other end plugs into the Arcnet board of the second computer.

For systems of three to four computers, a passive hub ($79) is needed. The passive hub has connectors for four cables. Cables from each one of the computers is connected to the hub. Figure 1 illustrates the setup.

The passive hub simply connects the four cables. The cable length from one computer to another when connected through a passive hub is 200 feet. For longer distances and if there are more than four computers connected, the more expensive active hub is needed ($999).

The active hub, which requires an AC outlet, amplifies the signals and allows you to connect more computers together at greater distances. The active hub has eight connectors which may be connected directly to computers, passive hubs, or other active hubs (two passive hubs cannot be connected directly together — the signal would suffer too much). The total cable length between a computer and an active hub is limited to 2,000 feet. Figure 2 shows how this is done.

A system with one active hub can have up to 24 computers before another active hub is needed. This is accomplished by connecting a passive hub to each one of the eight connectors on the active hub, and having three computers attached to each passive hub (the fourth connector on the passive hub is the one connected to the active hub).

While you are out buying all your ViaNet boards and cables, buy some extra memory for your computers. The ViaNet software takes about 128K of RAM. If you have a program that requires 256K, you'll need 384K to run it with ViaNet.

The Software

The software from ViaNetix, Inc. is the real gem of this network system. Packaged with each board you purchase are two diskettes. One contains the ViaNet software for your particular computer, and the other has text files of planning forms which you may print and use when designing your system.

There are two major components to the ViaNet software. One is the `NETDRV R.SYS` device driver which you install in your `CONFIG.SYS` file (see MS-DOSsier in this issue for more information on system configuration). The other, which is loaded from the MS-DOS prompt, is the actual network driver. Once the software is loaded, it resides in the background. The computer returns to the normal MS-DOS prompt, and the only indication you have that the network is running is when you are able to access devices on other computers.

In addition to the basic networking drivers, the ViaNet system includes utility software that allows you to set up groups of users and individual user passwords, software to spool print requests from users to one or more printers, software to change the protection status of individual files (files can be made so that only certain users

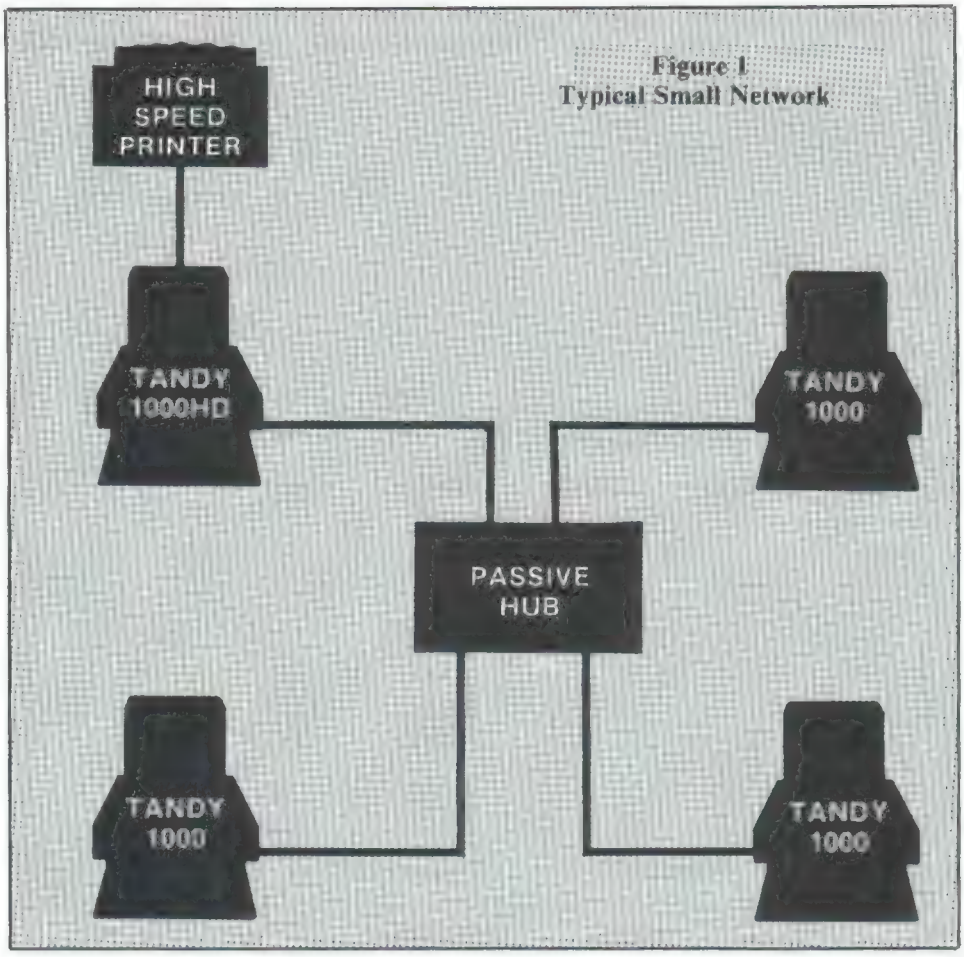


Figure 1
Typical Small Network

have access to them) and software to set up "macros" (more on macros follows).

Drive Z:
The ViaNet Connection

A typical single-user Tandy 1000 might have a floppy Drive A:, a hard disk C:, and perhaps a printer PRN:. When your computer is part of a ViaNet system, you have a new drive — Drive Z:.

Drive Z: is your software connection to ViaNet. All remote computers are accessed through Drive Z:. The root directory of Drive Z: contains a directory for each computer connected to the system. The names of the computers can be anything you want: Generic as "PC1" and "PC2" or personal such as "MARLA" and "WARREN." If you did a directory of the root of drive Z:, you would see something like Figure 3.

In Figure 3, you see that there are three computers connected to the network. We'll assume that we are PC1 and the other two are PC2 and PC3.

If we changed to the PC2 directory of Drive Z: and did a directory, we might see something like that in Figure 4.

As you can see in Figure 4, there are three files named NUL, CON, and PRN and two directories named A and C. NUL is the "nowhere" device on that machine — if you send something there, it goes nowhere. CON is the console — the keyboard and the display. PRN is the printer connected to that com-

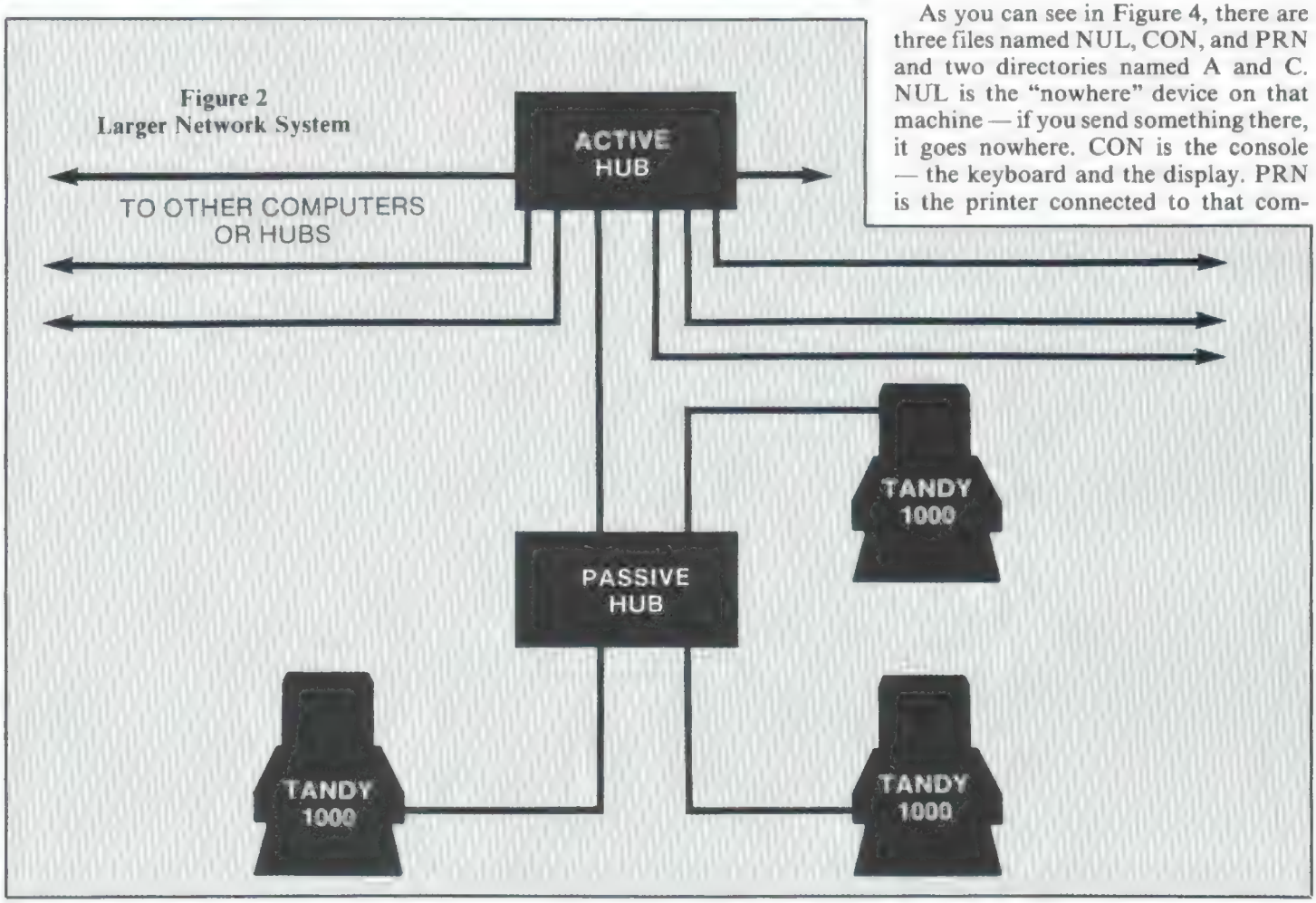


Figure 2
Larger Network System

Figure 3

```
 Volume in drive Z has no label
 Directory of  Z:\

PC1          <DIR>
PC2          <DIR>
PC3          <DIR>
        3 File(s) 125821440 bytes free
```

Figure 4

```
 Volume in drive Z has no label
 Directory of  Z:\PC2

NUL                 0
CON                 0
PRN                 0
A            <DIR>
C            <DIR>
        5 File(s) 125821440 bytes free
```

Figure 5

```
 Volume in drive Z has no label
 Directory of  Z:\PC2\C

COMMAND  COM    15957  10-20-84   1:00p
ANSI     SYS     4399  10-20-84   1:00p
CHKDSK   COM     6468  10-20-84   1:00p
DEBUG    COM    12223  10-20-84   1:00p
DISKCOMP EXE     5504  10-20-84   1:00p
DISKCOPY COM     1409  10-20-84   1:00p
DISKTYPE COM      563  10-20-84   1:00p
EDLIN    COM     8080  10-20-84   1:00p
EXE2BIN  EXE     1649  10-20-84   1:00p
FC       EXE     2585  10-20-84   1:00p
FIND     EXE     6331  10-20-84   1:00p
FORMAT   COM     6756  10-20-84   1:00p
GRAPHICS COM      959  10-20-84   1:00p
KEYCNVRT SYS      202  10-20-84   1:00p
LF       COM      135  10-20-84   1:00p
LIB      EXE    23168  10-20-84   1:00p
LINK     EXE    42330  10-20-84   1:00p
LPDRVR   SYS     2694  10-20-84   1:00p
LPINST   EXE     3072  10-20-84   1:00p
MODE     EXE    32910  10-20-84   1:00p
MORE     COM     4364  10-20-84   1:00p
ONEDISK  EXE      786  10-20-84   1:00p
PRINT    COM     4506  10-20-84   1:00p
RECOVER  COM     2295  10-20-84   1:00p
SORT     EXE     1632  10-20-84   1:00p
SYS      COM      922  10-20-84   1:00p
TREE     COM     1472  10-20-84   1:00p
BASIC    EXE    70000  10-20-84   1:00p
COPYDOS  BAT      734  10-20-84   1:00p
       29 File(s) 125821440 bytes free
```

puter. The directory called A is the disk inserted in Drive A of PC2 and C is the directory of the hard disk Drive C:.

This is simply a list of the devices and disk drives on PC2 made available to us. In a typical application, the floppy disk Drive A: on each computer would normally be private and only accessible to the person using that computer.

As we go into this directory structure to directory Z:\PC2\C — Drive C: on the computer named PC2. The directory might look something like Figure 5.

Figure 5 shows the directory of the hard disk Drive C: in PC2, but it was done from another computer. Instead of accessing it as Drive C: as we would if we were at the computer, we accessed it as Z:\PC2\C which can be thought of as "go through ViaNet to PC2 and to its Drive C." Any sub-directories of the drive would just be a continuation of the path Z:\PC2\C.

Macros

Mastering the art of remembering the path to the devices you want is not always easy — especially when many of the users on the system are computer users and not computer experts. Another problem is that most software will not recognize directory paths.

Fortunately, ViaNet software provides a wonderful program, MACASGN, which allows you to rename these paths to whatever suits your fancy. The path Z:\PC2\C, for example, could be renamed D:. D: would then become just another drive at your computer and could be used as such.

MACASGN can also be used to tell ViaNet where certain programs reside. If, for example, you have a program called MAILLIST which is always on the hard disk Drive C: of PC2, you could define a macro to say that "MAILLIST = Z:\PC2\C\MAILLIST." No matter which drive you're on and which directory you're in, if you ask for MAILLIST, ViaNet will help MS-DOS find it.

MACASGN is a utility that is very useful even when you are not concerned with using the network, but when you are accessing files and devices across the net, it is invaluable.

The system administrator, the person designated to setup and maintain the network, can create batch files and macros to make the system transparent and very easy to operate. As far as any user is concerned, the hard disk drives and printers are connected directly to his or her own computer.

Our Experiences with ViaNet

With the help of the good people at Tandy Center, we were able to get our ViaNet system up and running a little early. Using preliminary versions of the ViaNet software, we set up a system of four computers.

Computer one, a Tandy 1000, was equipped with a PBJ multifunction board with 512K and two 15M hard disk drives. The second computer, a Tandy 2000HD, had 512K RAM. A Tandy 1200HD 512K made up the third computer, and a Tandy 1000 with 384K and a Tallgrass tape backup system was computer number four.

We have been using *filePro 16* databases to keep track of products sent in for review, article and program submissions for publication, and an index for SOFT SECTOR, one of the other magazines published here.

The databases were all located on one of the hard disk drives connected to the first Tandy 1000. A series of *filePro 16* menus were set up to make it easy for anyone at any computer to call up any of the files. Since *filePro 16* does not currently use our multi-user installation, problems could arise if more than one person were working on the same file at the same time — and some did.

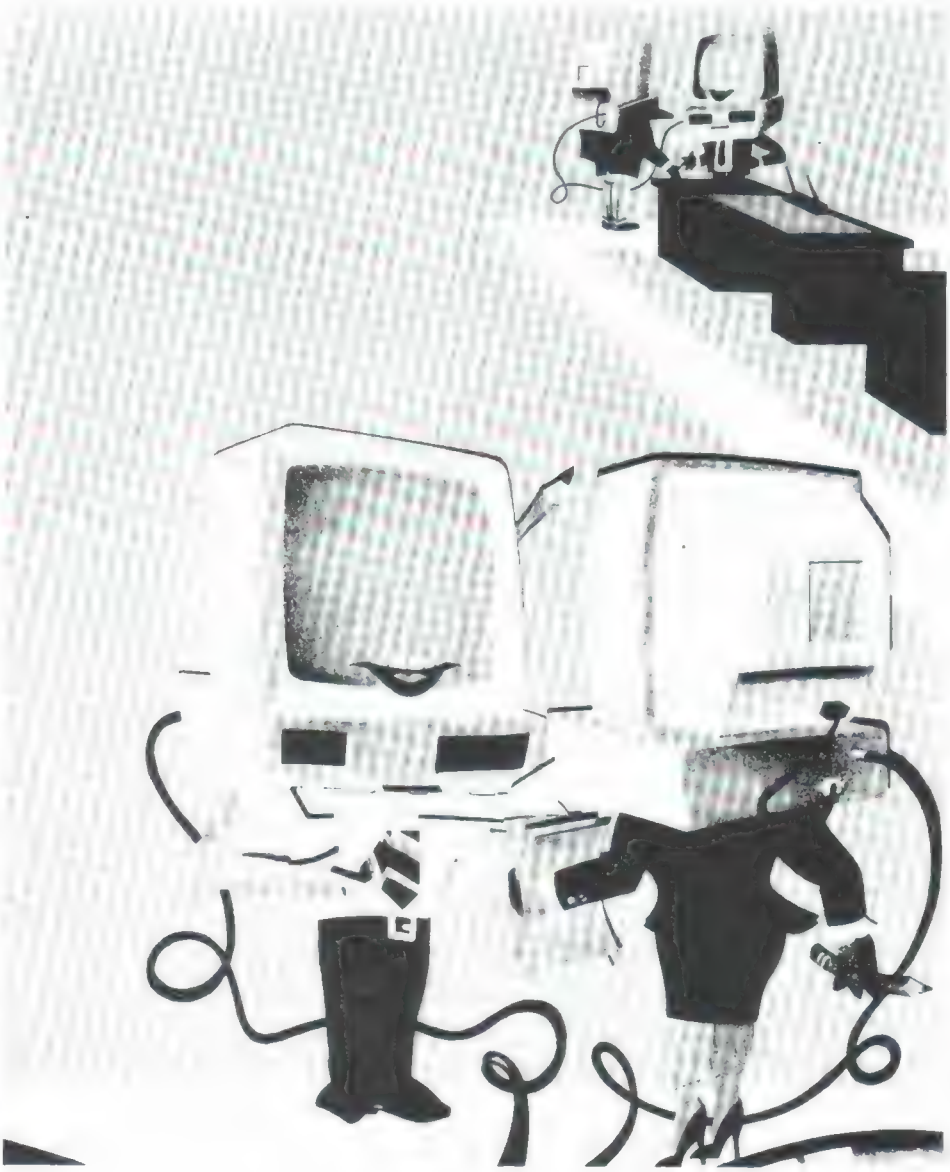
It was never a problem for more than one user to just look at a file, but when two people were trying to add to or update the file, conflicts existed. In order for software to work properly in a multi-user environment it must do two things. First, it must lock out a particular record from other users when you are updating that record. Secondly, it must be smart enough to wait if a record is currently locked by another user. ViaNet provides MS-DOS 3.00 compatible machine language calls for software that does behave this way.

We set up a simple internal mail system on the system with extensive use of batch files. Instead of practical uses, we found ourselves sending cute little notes across the office — the adult version of passing notes in the classroom.

All in all, the system worked wonderfully. We did have problems with some of the people resting or turning off their computers while others were trying to access hard disk drives connected to them — a practice we soon learned to avoid.

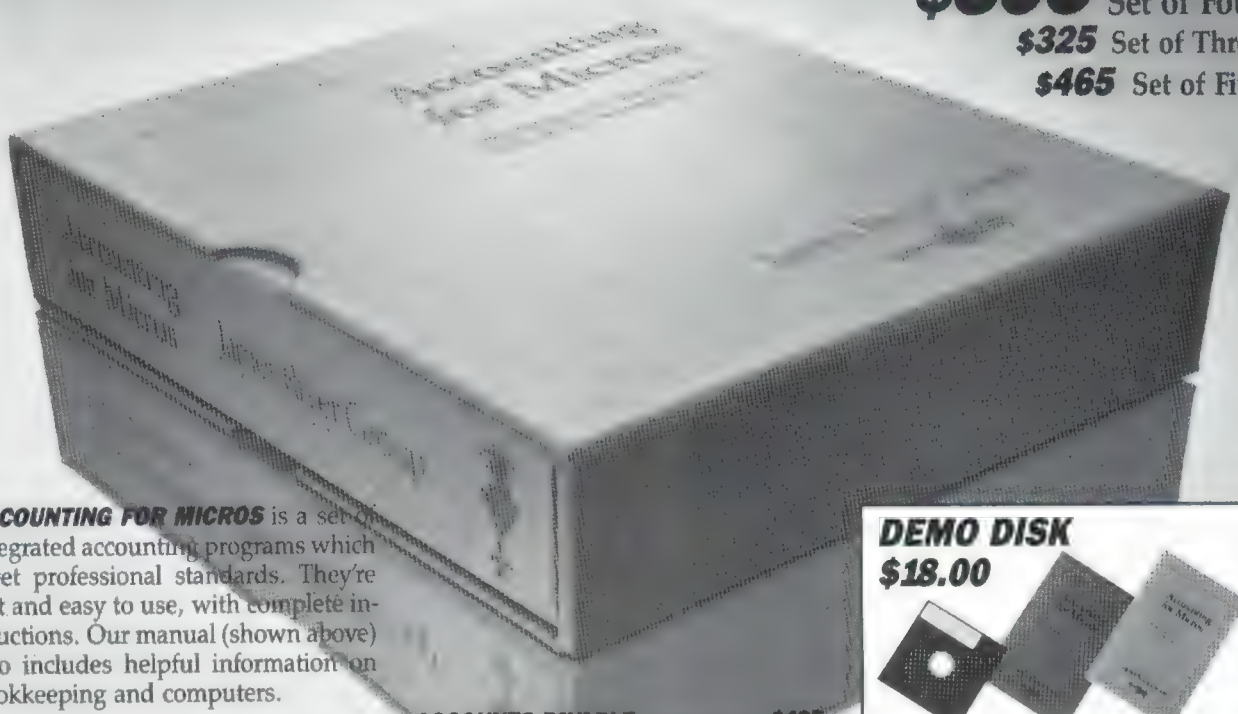
In just the short time that we have had ViaNet, it has become an essential part of our operation. The applications we are now doing on the network were once on a single machine. People had to almost literally take a number to use the computer or the desk where it was sitting — which happened to be mine! Now, since everyone involved with those applications has a computer at her desk, they can use the system whenever they need.

As we prepare to move into our new offices in the Falsoft Building, we see the network growing to a much grander scale. We plan to use it extensively for word processing — as articles are finished they can be sent directly to the typesetting department. The amount of paper that passes across desks can be reduced a bit by using the system for inter-office memorandums and messages. Certain departments will have applications directly related to their needs. It is even possible that PCM's subscriptions, which are now kept on a Tandy 6000 system, can be moved to the network someday. We have a lot of plans. PCM



ACCOUNTING FOR MICROS

$395 Set of Four
$325 Set of Three
$465 Set of Five



ACCOUNTING FOR MICROS is a set of integrated accounting programs which meet professional standards. They're fast and easy to use, with complete instructions. Our manual (shown above) also includes helpful information on bookkeeping and computers.

GENERAL LEDGER $125
Allows up to 1,000 accounts & 1,000 transactions/month. Retains mo/end balances for Last year, This Year and Forecast. Includes Cash Disbursements, Cash Receipts and General Journals. Reports include Balance Sheet, Income Statement, Annual Summaries and Journal Reports.

ACCOUNTS RECEIVABLE $125
Allows up to 2,500 customers and 1,000 invoices per month. Invoicing can access Inventory Module. Keeps customer names and addresses. Invoice prints on plain paper or any pre-printed form. Statements can be printed at any time.

INVENTORY $125
Allows up to 4,000 parts. Keeps 3 month history of unit sales as well as year to date. With AR, can be used as point of sale system (prints invoices, handles cash). Reports include Inventory Value and Stock Report, Internal and Customer Price List.

ACCOUNTS PAYABLE $125
Allows up to 500 vendors and 600 invoices/mo. Records invoices and handwritten checks. Prints computer checks on any pre-printed form. Keeps vendor names and addresses.

PAYROLL $125
Will handle up to 100 employees with six deductions per employee. Deductions may be determined as fixed dollar amounts or percentages, or referred to a table for automatic look-up. Tax tables are easily entered, or purchased separately.

SET OF FIVE $465
SET OF FOUR $395
SET OF THREE $325

RUN ON MOST CPM AND MSDOS

Apple CPM	*IBM PC,XT,PC jr,AT*	*Sanyo (all)*
Columbia	*Kaypro (all)*	*Tandy (all)*
Compaq	*Morrow (all)*	*TeleVideo*
Corona	*Osborne (all)*	*Zenith 100 & 150*
Eagle (all)	*Panasonic*	*8" CPM*
Epson QX-10	*Radio Shack CPM*	*Other compatibles*

DEMO DISK
$18.00

Try all 5 programs above (GL, AR, AP, IN, PR). Order our DEMO DISK for $18.00 (includes shipping). Condensed versions of the programs give you the "feel" of data entry and access. Includes sample reports and instructions. Specify machine.

TRASHMAN (NEW) $145
The "Catch-All" program. Files any type of information for quick access. Name or subject oriented with 15 lines of notes per name. Use TMAN as a mailing list, filing system, notebook, etc. Can be used alone or with data from our other programs.
Try ***TRASHMAN (only) DEMO $16***

HOW TO ORDER: Please specify machine and disk format. You can pay by check, by VISA or MasterCard (we need your expiration date and card number), or by UPS COD (add $2.50 COD charge). Our price includes shipping. Minnesota residents, add 6% sales tax). We ship most orders the same day.
or ***ORDER BY PHONE: 612-339-2521***

James River Group
INCORPORATED

(612) 339-2521

125 North First Street
Minneapolis, MN 55401

Herein, the author develops subroutines to allow BASIC to access the most useful MS-DOS functions

Accessing MS-DOS Commands from BASIC

By Robert D. Covington

Have you ever wanted to make a directory in BASIC? Have you ever wanted to change the current directory, change the default drive, or determine the amount of available space on a disk? If so, you are probably well aware of BASIC's lack of support of MS-DOS.

In this installment of "Subroutine City", a set of subroutines for BASIC will be described to allow BASIC to access most of MS-DOS's disk management, file management, and file I/O functions.

MS-DOS's Internal Structure

The MS-DOS disk operating system used in the Model 1000-2000 is divided into three seperate sections. The first section, the basic input/output system (BIOS for short), is responsible for controlling all of the computer's hardware accessed by MS-DOS. The routines in the BIOS are used to print a character on the screen, update the internal time clock, get a character from the keyboard, etc. In general, the BIOS is the interface between the MS-DOS system and the hardware.

The second section of MS-DOS contains the actual routines for supporting the disk operating system. This section is used for managing files, loading programs, managing file directories, etc. Generally, most of the functions used by MS-DOS are accessed through this section.

The third and final section of MS-DOS, the command processor, is responsible for interfacing MS-DOS to the user. The command processor controls all of MS-DOS's commands (DIR, REN, DEL, etc.), device redirection and piping, batch file processing, etc.

To help you better understand the functions of these three sections of MS-DOS, let's take a look at what occurs when BASIC is loaded off disk from command level. First, the user enters BASIC followed by an enter at command level. Next, the command processor looks at what was entered and checks to see if it is a legal MS-DOS internal command. Since in this case the entered text was not a command, the command processor assumes that a request was made to load a program. After processing any pipes or device redirection, the command processor tells MS-DOS to attempt to load the program. MS-DOS then checks to see if the selected disk contains the file that was requested. If the file is not on the disk, MS-DOS returns with an error message, the command processor prints the textual equivalent of the error, and control is returned to the command processor. If the file was found, MS-DOS uses the BIOS to load the program from disk to memory. Then, MS-DOS computes all the relocatable code and transfers control to the loaded program.

When BASIC is loaded into memory, its only access to MS-DOS is through the BIOS or the actual MS-DOS system. The command processor that loaded BASIC is temporarily halted and thus can not be used by BASIC. As a matter of fact, you could consider the actual BASIC interpreter as a new command processor; interfacing the computer user to MS-DOS. When BASIC is "in control," it uses MS-DOS and its BIOS for keyboard scanning, file I/O, directory management, RS-232 support, etc. Unfortunately, BASIC uses MS-DOS's functions internally and does not normally allow the user to make direct use of the functions inside MS-DOS.

MS-DOS Access Subroutine

Thanks to BASIC's CALL instruction, a machine language subroutine can be created to access all of MS-DOS's internal functions inside a BASIC program. This means that BASIC programs will now be able to make use of the many functions supported by MS-DOS. In addition, this same subroutine will allow BASIC to go down one more step and directly access the BIOS (more on this in future installments).

Program 1 contains the source for a universal MS-DOS/BIOS function call subroutine for BASIC. This machine language subroutine loads the 8088/80186's eight main user registers with eight BASIC variables respectively. The subroutine expects BASIC's variables to be integers and to correspond to the registers in the order AX, BX, CX, DX, BP, SI, DI, status word. Then, the subroutine causes a forced interrupt to one of the functions in MS-DOS or the BIOS. When the function is complete, the subroutine loads the contents of the updated user registers back into the BASIC variables used before. (See last month's "Subroutine City" for more information on machine language subroutines and how variables are passed to them.)

In Program 2, the machine language program has been integrated into a BASIC program. Lines 10-20 set up and load the assembled version of Program 1. To better adapt this machine language subroutine to BASIC, the assembled subroutine is converted into DATA statements in lines 50000 through 50002. These data statments are loaded

(Bob Covington has been a computer programmer and consultant for the past six years, most recently focusing his attention on both the Model 100 and the 2000. He is also a technical writer and editor. Bob can be contacted at P.O. Box 37007, St. Louis, MO 63141.)

Listing 1

```
        PAGE    60,80
        TITLE   MSDOS Function call Accesser ** By Robert D. Covington
CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE
        PUBLIC  BIOS
BIOS    PROC    FAR
START:
        PUSH    DS                  ;MOV ES,DS
        POP     ES
        MOV     BP,SP               ;Make BP base pointer of stack
        MOV     BX,[BP+4]           ;Get descriptor for SW variable
        PUSH    [BX]                ;Move SW,[BX]
        POPF
        MOV     BX,[BP+6]           ;Get descriptor for DI variable
        MOV     DI,[BX]             ;Put variable in DI
        MOV     BX,[BP+8]           ;Get descriptor for SI variable
        MOV     SI,[BX]             ;Put variable in SI
        MOV     BX,[BP+12]          ;Get descriptor for DX variable
        MOV     DX,[BX]             ;Put variable in DX
        MOV     BX,[BP+14]          ;Get descriptor for CX variable
        MOV     CX,[BX]             ;Put variable in CX
```


Why Handle Files One at a Time? Let XOUT/XIN from BKI do the work for you.

Eliminate tedious (and fault prone) load/save/kill and format procedures on your MODEL 100. XOUT/XIN programs automatically create/read and display RAM and external directories. Resulting free space is displayed as files are selected/deselected to aid in memory management. Menus and prompts provide a safe, easy-to-use file management utility—an amazing productivity tool.

Use XOUT.BA to send multiple copies of selected memory files to the tape (**Save**); or send them directly (**List**), or via a formatter (**4mat**), to either the printer or the screen. Selected files can simply be removed with the **Kill** command and confirmation. An external directory is automatically created for each copy of a file set that is saved (containing creation date and time, and an ordered list of names and file sizes). Files saved on tape, including the directory, may also be loaded one-by-one by BASIC or TEXT, in the usual way.

Use XIN.BA to automatically **Load** selected tape files back into memory; or send them directly (**List**), or via a formatter (**4mat**), to the printer or the screen.

Typical XOUT.BA Menu

Typical XIN.BA Menu

PCM CERTIFICATION SEAL

– XOUT.BA –

- Displays memory contents in memory address order
- As files are highlighted memory statistics are changed to reflect values if **Kill** selected.
- BA files are stored on tape as standard tokenized files.

– XIN.BA –

- Displays tape contents in tape position order
- As files are highlighted memory statistics are changed to reflect values if **Load** selected.
- If the file name exists in both the tape and memory directories, the difference (+, -,0) in size is shown.

- Label line alternatives with statistics line by toggling the label key
- Any combination of files can be highlighted by positioning the pointer and pressing space or enter. No action is taken until commanded by a function key, and then all highlighted files are affected.
- The **4mat** command invokes the BA4MAT.BA program to format BA files and the DO4MAT.BA program to format DO files.

Skeleton BA4MAT.BA and DO4MAT.BA programs are provided as examples for interfacing existing formatters. Full featured BA4MAT.BA and DO4MAT.BA programs, already interfaced with the XOUT/XIN programs, are offered as separately priced items. DO4MAT.BA and BA4MAT.BA format a single file if executed from the main menu or multiple files if executed via XOUT or XIN.

– DO4MAT.BA –

- Formats documents using embedded command descriptions for many features, including columns, underlining, **bold print**, filling and adjusting.
- Optional headers and footers may include the date, time and page number
- Specify printer make and model (Customizing directions given for printers not yet implemented.)

– BA4MAT.BA –

- Formats BA files so that the individual basic commands are listed one per line indented from the basic line number
- Additional indentation occurs to diagram the IF-THEN-ELSE structures in the program
- Pagination including the date, time and page number are provided.

XOUT.BA/XIN.BA
DO4MAT.BA (skeleton)
BA4MAT.BA (skeleton) 49.95
DO4MAT.BA (full featured) 39.95
BA4MAT.BA (full featured) 29.95

Buy two programs take $10 discount.
Buy all three take $20 discount.
MD residents add 5% Sales Tax.

All programs come on cassette with documentation.

Send Check or Money Order to:
BKI
PO Box 218
College Park, MD 20740
Questions and COD call (301) 345-9473

BKI

```
        MOV     BX,[BP+18]      ;Get descriptor for AX variable
        MOV     AX,[BX]         ;Put variable in AX
        MOV     BX,[BP+16]      ;Get descriptor for BX
        MOV     BX,[BX]         ;Put variable in BX
        MOV     BP,[BP+10]      ;Get descriptor for BP
        MOV     BP,[BP]         ;Put variable in BP
        INT     21H             ;Execute DOS function call
        PUSH    BP              ;Save BP for later
        PUSH    BX              ;Save BX for later
        MOV     BP,SP           ;Make BP base of stack
        MOV     BX,[BP+8]       ;Get descriptor for SW variable
        PUSHF                   ;MOV [BX],SW
        POP     [BX]
        MOV     BX,[BP+10]      ;Get descriptor for DI variable
        MOV     [BX],DI         ;Save DI in variable
        MOV     BX,[BP+12]      ;Get descriptor for SI variable
        MOV     [BX],SI         ;Save SI in variable
        MOV     BX,[BP+16]      ;Get descriptor for DX variable
        MOV     [BX],DX         ;Save DX in variable
        MOV     BX,[BP+18]      ;Get descriptor for CX variable
        MOV     [BX],CX         ;Save CX in variable
        MOV     BX,[BP+22]      ;Get descriptor for AX variable
        MOV     [BX],AX         ;Save AX in variable
        POP     BX              ;Restore BX
        MOV     BP,[BP+20]      ;Get descriptor for BX variable
        MOV     [BP],BX         ;Save BX in variable
        MOV     BP,[BP+14]      ;Get descriptor for BP variable
        POP     BX              ;Put old BP in BX
        MOV     [BP],BX         ;Save BP in variable
        RET     16              ;Return fixing for 8 parameters
BIOS    ENDP
CODE    ENDS
        END     START
```

Listing 2

```
1 ' *** MSDOS Function Subroutines for BASIC
2 ' *** By Robert D. Covington
10 DIM B$,AX,BX,CX,BP,SI,DI,SW
20 FOR X=1 TO 100:READ A:BIOS$=BIOS$+CHR$(A):NEXT:X=FRE(X):V=VARPTR(BIOS$):BIOS!
=PEEK(V+1)+PEEK(V+2)*256
25 CLS:PRINT"Program Installed...."
999 END
2000 ' *** Create File
2001 '  Entry:
2002 '          A$ = File name
2003 '  Exit:
2004 '        FH% = File Handle
2005 '          E = Error Status (0-no error)
2006 '
2010 AX%=15360:CX%=0:GOTO 3020
3000 ' *** Open already existing file
3001 '  Entry:
3002 '          A$ = File name
3003 '  Exit:
3004 '        FH% = File Handle
3005 '          E = Error Status (0-no error)
```

into the string BIOS$ by Line 20. By doing this, the machine language subroutine can be used on almost any MS-DOS system with almost any memory configuration.

Disk Management Subroutines

At this point, most of you are probably wondering what the capabilities of this wonderful machine language subroutine are. Let's take a look at some of the other subroutines contained in Program 2 that make use of this machine language subroutine.

The subroutine starting at Line 8000 sets the current default drive for MS-DOS to the drive named in A$. The default drive is the drive that MS-DOS uses to search for a file if no drive name is given in the file's pathname. For example, if you performed the following function:

```
A$="B":GOSUB 8000
```

all file accesses after the GOSUB would use drive B unless otherwise directed to by the pathname (example, A:TEXT.DAT). On return from the subroutine, the variable D contains the number of drives supported by the current BIOS and E contains the error status. If the error status is 0, no error occured. If the error status is 15, the drive selected is illegal.

If you wish to determine which drive is designated as the default drive, the subroutine starting at 9000 can be used. This subroutine returns with the drive name in A$ (A-E) and the drive number in D (A-0, B-1, etc.).

Probably one of the most useful functions for an entire disk that is omitted from GW-BASIC is a function that returns the amount of free space available on a disk. The subroutine starting at Line 10000 will not only find the amount of free space on a disk but also the total storage capacity of the disk. On entry to this routine, the variable D must contain the drive number you wish to obtain the free space information from. If D is equal to 0, the default drive is selected. If D is greater than 0, the drive number directly corresponds to the letter of the drive (example, 1-A, 2-B, 3-C, etc.). Upon return from this subroutine, F! contains the number of free bytes on the system and A! contains the total storage capacity of the disk. In addition to these variables, four other variables on the specifications of the drive are given (see lines 10004-10007). Since these commands are more technically oriented, I will not describe them in this article.

Directory Management Subroutines

Another important feature that is omitted on GW-BASIC is subdirectory support. Considering that MS-DOS's most powerful function is its tree directory structure, this is quite a loss.

To alleviate this problem, the machine language MS-DOS access subroutine can be used to tap into MS-DOS's powerful directory support.

On all of the directory management subroutines, the E register returns with the error status. If E is equal to 0, no error occurred. If E is greater than 0, the following errors occurred:

Value in E	Error
3	Path Not Found
5	Access Denied. Directory is not empty, legal, or is the current directory.
15	Invalid Drive

```
3006 '
3010 AX%=15618
3020 I=33:A$=A$+CHR$(0):GOSUB 32000:GOSUB 40000:IF (SW% AND 1)=0 THEN FH%=AX%:E=
0:ELSE E=AX%
3030 RETURN
4000 ' *** Close file
4001 '   Entry:
4002 '           FH% = File handle
4003 '   Exit:
4004 '             E = Error status (0-no error)
4010 BX%=FH%:AX%=15872
4020 I=33:GOSUB 40000:IF (SW% AND 1)=0 THEN E=0 ELSE E=AX%
4030 RETURN
5000 ' *** Move file Position
5001 '   Entry:
5002 '           FH% = File handle
5003 '            A! = File position (byte reletive to start of file)
5004 '   Exit:
5005 '             E = Error status (0-no error)
5006 '
5010 I=33:AX%=16896:BX%=FH%:GOSUB 30000:GOTO 4020
6000 ' *** Read data from file
6001 '   Entry:
6002 '           FH% - File handle
6003 '            L - Number of characters to read (LRL)
6004 '   Exit:
6005 '            A$ - Data read from file
6006 '             E - Error status (0-no error)
6007 '             L - # of characters actually read
```

16 Attempt to remove the current directory

The subroutine starting at 11000 will create a subdirectory with the pathname in A$. For example, if you executed an:

```
A$="BASIC\DATA":GOSUB 11000
```

the directory BASIC\DATA would be created on the current default disk.

The subroutine starting at 12000 deletes the subdirectory with the pathname in A$ from the disk. As at MS-DOS command level, the directory must be completely empty before the directory can be removed.

The subroutine starting at 13000 changes the current default directory path to the pathname in A$. For example, an:

```
A$="\":GOSUB 13000
```

makes the root system directory the current default directory just as if a CD\ were entered at MS-DOS command level.

The subroutine starting at 14000 returns the current default directory path in A$. On entry to this routine, D must contain the drive number for the drive you wish to retrieve the pathname from. If D is loaded with a 0, the current default drive is used. If D is greater than 0, the drive number directly corresponds to the letter of the drive (example, 1 - A, 2 - B, etc.). This function in conjunction with the routine at 13000 is very useful in moving up or down directory levels.

The subroutine starting at 15000 renames the file with the pathname in A$ to the pathname in B$. Best of all, this function not only allows the file name to be changed, but it also allows a file to be moved to another directory on the same drive. For example, if an:

```
A$="BASIC\DATA.TXT":B$="DATA\
DATA.TXT":GOSUB 15000
```

(a one-line entry on your computer) were executed, the file DATA.TXT would be moved from the BASIC directory to the DATA directory. Unfortunately, this function does not support moving a file between disk drives. Because of this, the paths in A$ and B$ must refer to files on the same drive.

File I/O Subroutines

You are probably wondering why we would need to access MS-DOS's file I/O subroutines when BASIC has such powerful file manipulation instructions. Well, BASIC's file I/O instructions work fine in all but one case; when a file is being manipulated on a byte-per-byte basis.

One of the most annoying limits imposed by BASIC is the 32767 record limit on a GET or PUT instruction. This means that if you want to randomly access a file with a logical record length of one byte, a file only 32K long can be easily accessed. This limitation has resulted in quite a few problems in many programs I have written.

Where BASIC suffers, MS-DOS triumphs. MS-DOS's file I/O functions allow random access of files up to 4,294,967,296 bytes long. Since I doubt that too many people will ever exceed that limit, I think I can safely say that MS-DOS's file I/O functions can manipulate any file on a byte-per-byte basis.

Like BASIC, MS-DOS's file I/O functions require that a file be opened

```
6008 '
6010 AX%=16128:A$=SPACE$(L)
6020 BX%=FH%:CX%=L
6030 I=33:GOSUB 32000:GOSUB 40000:IF (SW% AND 1)=0 THEN L=AX%:E=0 ELSE E=AX%
6040 RETURN
7000 '  *** Write data to file
7001 '  Entry:
7002 '          FH% - File handle
7003 '           A$ - Data to write to file
7004 '  Exit:
7005 '           L - Length of data written
7006 '           E - Error Status (0-no error)
7010 AX%=16385:L=LEN(A$):GOTO 6020
8000 '  *** Set new default drive
8001 '  Entry:
8002 '           A$ - Drive Name (ex. AX="A")
8003 '  Exit:
8004 '           D - Number of logical drives on line
8005 '           E - Error Status (0-no error)
8006 '
8010 AX%=3584:E=0:DX%=(ASC(A$) AND 223)-65:IF DX<0 OR DX>25 THEN E=15:RETURN
8015 I=33:GOSUB 40000:D=CVI(LEFT$(MKI$(AX%),1)+CHR$(0)):RETURN
9000 '  *** Get current default drive
9001 '  Exit:
9002 '          A$ - Drive Name
9003 '           D - Drive number (0-A, 1-B, etc...)
9004 '
9010 AX%=6400:GOSUB 8015:A$=CHR$(65+D):RETURN
10000 '  *** Get ammount of free space on disk
```

```
10001 '   Entry:
10002 '           D - Drive (0-default, 1=A, 2=B, etc.)
10003 '   Exit:
10004 '          BX% - Number of free granules
10005 '          DX% - Number of total granules on drive
10006 '          CX% - Bytes per sector
10007 '          AX% - Sectors per granule
10008 '           F! - Number of free bytes on system
10009 '           A! - Total storage capacity of drive in bytes
10010 '
10020 AX%=13824:DX%=D:I=33:GOSUB 40000:IF AX%=-1 THEN E=15:RETURN
10030 F!=BX%*CX%*AX%:A!=DX%*CX%*AX%:RETURN
11000 ' *** Make Subdirectory
11001 '   Entry:
11002 '          A$ - Path of new subdirectory
11003 '   Exit:
11004 '           E - Error status (0-no error)
11010 AX%=14592
11020 A$=A$+CHR$(0):I=33:GOSUB 32000:GOTO 4020
12000 ' *** Remove Subdirectory
12001 '   Entry:
12002 '          A$ - Path of subdirectory to delete (must contain no files)
12003 '   Exit:
12004 '           E - Error status (0-no error)
12005 '
12010 AX%=14848:GOTO 11020
13000 ' *** Change current subdirectory
13001 '   Entry:
13002 '          A$ - Path of subdirectory to change to
13003 '   Exit:
13004 '           E - Error status (0-no error)
13005 '
13010 AX%=15104:GOTO 11020
14000 ' *** Get Name of current directory path
14001 '   Entry:
14002 '           D - Drive number (0-default drive, 1-A, 2-B, etc...)
14003 '   Exit:
14004 '          A$ - 64 character string of current path
14005 '           E - Error Status (0-no error)
14006 '
14010 AX%=18176:A$=SPACE$(64):GOSUB 32000:SI%=DX%:DX%=D:GOTO 4020
15000 ' *** Move/Rename a directory entry on a single drive
15001 '   Entry:
15002 '          A$ - Current pathname of file
15003 '          B$ - New pathname for file
15004 '   Exit:
15005 '           E - Error status
15010 A$=A$+CHR$(0):B$=B$+CHR$(0):AX%=22016:SWAP A$,B$:GOSUB 32000:DI%=DX%:SWAP
A$,B$:GOSUB 32000:GOTO 4020
30000 ' *** Convert single precision to double word
30001 '   Entry:
30002 '           A! - Single Precision Integer
30003 '   Exit:
30004 '          CX% - Most significant word
30005 '          DX% - Least significant word
30006 '
30010 A1=A!/65536!:A2!=(A1!-INT(A1!))*65536!:A!=A1!:GOSUB 31000:CX%=A%:A!=INT(A2
!):GOSUB 31000:DX%=A%:RETURN
```

before it can be accessed. When a file is opened, MS-DOS stores all the data that is needed to manipulate the file in one of its internal buffers. The number of buffers that MS-DOS can have open at one time is specified by the FILES parameter in the CONFIG.SYS file loaded when the system is booted (see Appendix C of the MS-DOS user manual for more information on CONFIG.SYS). MS-DOS allows up to 99 files to be allocated and opened at the same time. This is an especially nice improvement over BASIC's 15 limit.

When MS-DOS returns from opening a file, a 16-bit number called a file handle is returned to the calling program. This file handle is used to describe a file in much the same way that a file buffer number does in BASIC. Whenever any file I/O is performed on a file, the file handle tells MS-DOS which of the open files is to be manipulated. (See "File I/O Techniques for Your Model 1000, 1200 and 2000" in the March 1985 issue of PCM for more information on file handles and MS-DOS's internal file I/O structure.)

In MS-DOS, there are two ways to open a file; create and open. Create, as the name implies, is used to create and open a new file or to start an old file from scratch (deleting all the information stored in the file). Open is used to open an already existing file for further access and modifications. Open cannot be used on a file that has not been created previously.

The subroutines starting in lines 2000 and 3000 respectively, create or open a file. On entry to each of these subroutines, A$ must contain the pathname of the file to open or create (subdirectories can be used in the path if desired). Upon return from these subroutines, FH% contains the file handle for the file and E contains the error status. If E is equal to 0, no error occurred. If E is greater than 0, the E corresponds to the folling errors:

Value in E	Error
2	File Not Found
3	Invalid Path
4	Too many files open. More files than allocated by the FILES parameter of CONFIG.SYS were opened.
5	Access denied.

The subroutine starting at Line 4000 closes the file with the file handle in FH%. Like BASIC's CLOSE, this subroutine writes any remaining data to disk, updates the directory, and nullifies the file open. On return from this routine, E contains the error status of the operation. If E is equal to 0, no error occurred. If E is equal to 6, the file handle specified is invalid.

The subroutine starting at Line 5000 directs MS-DOS to move its current file position pointer to any byte in the file. This function allows random access to any open file on a byte-per-byte basis. On entry to this subroutine, FH% must contain the file handle of the file to position and A! must contain the position relative to the start of the file to move to. On return from this subroutine, E contains the error status of the operation. The error status returned is identical to that of the previous subroutine.

The subroutine starting at Line 6000 reads L number of characters from an open file. On entry, FH% must contain the file handle of the file to read from and L must contain the number of characters to read. On return, A$ contains the text read from the file, L contains the number of characters

```
31000 ' *** Convert Integer to signed integer
31001 '  Entry:
31002 '           A! - Integer (0-65535)
31003 '  Exit:
31004 '           A% - Signed Integer (-32768 - 32767)
31005 '
31010 IF A!>32767 THEN A%=A!-65536! ELSE A%=A!
31015 RETURN
32000 ' *** Find address for string
32001 '  Entry:
32002 '           A$ - String to find address of text
32003 '  Exit:
32004 '          DX% - Address of text in string
32005 '
32010 V!=VARPTR(A$):A!=PEEK(V!+1)+PEEK(V!+2)*256:GOSUB 31000:DX%=A%:RETURN
40000 ' *** Call DOS function
40001 ' I - Interrupt Number
40002 ' AX%, BX%, CX%, DX%, BP%, SI%, and DI% - Registers passed to and from DOS
40003 ' SW% - Status word
40004 '
40010 MID$(BIOS$,48,1)=CHR$(I):CALL BIOS!(AX%,BX%,CX%,DX%,BP%,SI%,DI%,SW%):RETUR
N
50000 ' Program:BIOS        Length: 100 bytes
50001 DATA 30,7,139,236,139,94,4,255,55,157,139,94,6,139,63,139,94,8,139,55,139,
94,12,139,23,139,94,14,139,15,139,94,18,139,7,139,94,16,139,31,139,110,10,139,11
0,0,205,33,85,83,139,236,139,94,8,156,143,7,139,94,10,137,63,139,94,12
50002 DATA 137,55,139,94,16,137,23,139,94,18,137,15,139,94,22,137,7,91,139,110,2
0,137,94,0,139,110,14,91,137,94,0,202,16,0
65000 SAVE"bios
```

actually read, and E contains the error status. If L is less than the number of characters requested, the end of the file has been reached. The error status returned is identical to that of the close subroutine. After each read instruction, the file pointer is automatically incremented by the number of characters read.

The subroutine starting at Line 7000 writes the data in A$ to disk at the current file position. On entry, A$ contains the data to write and FH% contains the file handle of the file to write to. If A$ contains a null string (A$=""), the end of file for the current file is set to the current file position. For example, if the file position is set to byte 100 of a 1000-byte file and an:

```
A$="":GOSUB 7000
```

were executed, the file would shrink to 100 bytes. All the data in bytes 101 through 1000 in the file would be deleted and deallocated by the system. On exit from this subroutine, L contains the number of characters actually written and E contains the error status. The error status returned is identical to that of the close subroutine.

Miscellaneous Subroutines

The subroutine starting at 30000 converts the single precision integer in A! into the double word in CX% and DX%. This function is used primarily by the file position subroutine to calculate the 32-bit file position.

The subroutine starting at 31000 converts the integer in A! to the signed integer in A%. In GW-BASIC, all integer variables have the ability of having a positive or negative sign. To allow this, the last bit in the integer, bit 15, is used to signify the sign of the number. In machine language, however, all integers are expected be a full 16 bits wide. To alleviate this incompatibility problem, this subroutine uses the sign bit (bit 15) of the integer variable A% as the 16th bit of the integer.

The subroutine starting at 32000 determines the start address of the data in the string variable A$. On exit from this routine, DX% contains the start address of A$ in BASIC's data segment.

Using Program 2 in Other Programs

When merging the routines in Program 2 with your applications program, a few precautions must be taken. First, lines 10-20 must always be the first lines executed by any program Program 2 is integrated into. Second, if the applications program uses its own DATA statements, the DATA in lines 50000-50002 must be moved so they are the first data lines read by the system.

In general, none of the subroutines in Program 2 should be deleted when integrating them into an applications program. Most of the subroutines in Program 2 use parts of other subroutines to eliminate any redundant program sections. Since most of the subroutines in Program 2 are very short, I would suggest leaving them in your programs even if you are not going to use them. If memory is tight, I suggest deleting the many REMarks from Program 2.

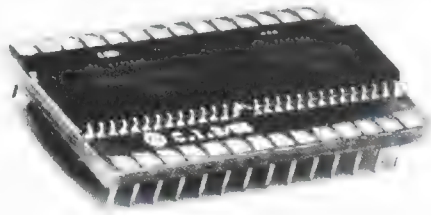
Conclusion

The subroutines in this month's installment of "Subroutine City" give BASIC many of the most needed functions accessible at MS-DOS command level.

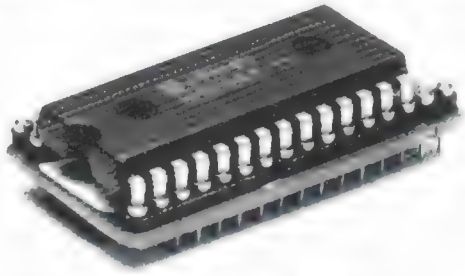
In the next installment, the MS-DOS access machine language subroutine will be used again to allow access to even more functions in MS-DOS that were normally restricted from BASIC.

PCM

24K Expansion RAM
For the Model 200
$150. each **$275.** for two



8K Expansion RAM
For the Model 100 & 8201
$35. each **$99.** set of 3



These Modules are easily user installable and they feature:

- Low Power CMOS Static RAM
- 30 Day Satisfaction Full Refund Policy
- High Quality Sockets
- Instructions Include Test Program Listing
- One Year Warranty

M/C, VISA, Check or Money Order
Shipping-Add $1.50 UPS Ground or $4.00 2nd Day Air.
CA Residents add 6% Sales Tax

Cryptronics, Inc.
11711 Coley River Circle, Suite 7
Fountain Valley, CA 92708
Phone: (714) 540-1174

Prevent data entry errors with these BASIC subroutines

A Better Input Processor

By Bill Qualls

Input processors appear frequently in microcomputer literature. The concept behind the input processor is simple: to isolate requests for entry and editing of entered data into a common multipurpose subroutine, usually accessed by a GOSUB statement.

Input processors represent an expansion on the subroutine concept. Subroutines can greatly simplify new program development. Not only can standard edits be performed, such as date validation and checking for numeric fields, but the input data can be returned to the main program in a standardized format. For example, why code this:

```
2130 INPUT "Want to enter more"; YN$
2140 IF YN$="YES" OR YN$="yes" OR YN$="Y"
     OR YN$="y" THEN 2000
2150 IF YN$="NO" OR YN$="no" OR YN$="N"
     OR YN$="n" THEN END
2160 PRINT "<ERROR> Enter YES or NO"
     : GOTO 2130
```

(Bill Qualls is an assembly language programmer and project leader at May & Speh, Inc., Midwest computer service bureau. Bill has a bachelor's degree in accounting and in June completed his MBA. He is also a computer's merit badge counselor for local area Boy Scouts.)

when instead you can call a subroutine which will allow the user to enter "yes/no" in any of the above formats, but will always return 'Y' or 'N.' This is much easier:

```
500 LINEINPUT YN$
510 IF YN$="YES" OR YN$="yes" OR YN$="Y"
    OR YN$="y" THEN YN$="Y"
    : RETURN
520 IF YN$="NO" OR YN$="no" OR YN$="N"
    OR YN$="n" THEN YN$="N"
    : RETURN
530 PRINT "<ERROR> Enter YES or NO"
    : GOTO 500
2130 PRINT "Want to enter more?";
     : GOSUB 500
     : IF YN$="Y" THEN 2000 ELSE END
```

I have seen input processors in school, in magazines and in the work place. Those I have seen can be grouped into three classes: Multiple Routine, Columnar Orientation and Multiple Variable.

Multiple Routine

This is the better-than-nothing class of input processors. The prompts for the data are issued from the main body of the program, but the editing of the data is isolated into common modules. The type of data required dictates which routine is to be called. For example:

To receive ( ) data	GOSUB ( )	Return variable
String	200	ST$
Numeric	300	NM#
Date	400	DT$
YES/NO	500	YN$

```
2130 PRINT "Enter Gross Sales";
     : GOSUB 300
     : GS = NM#
```

Some programmers have taken a big step toward improving this style by using a variable to indicate which type of data is required and calling a "master" routine which then decides which subroutine should additionally be called. For example:

To receive ( ) data	Let TY$ =	GOSUB ( )	Return variable
String	"S"	200	ST$
Numeric	"N"	300	NM#
Date	"D"	400	DT$
YES/NO	"Y"	500	YN$

```
2130 PRINT "Enter Gross Sales"
     : TY$ = "Y"
     : GOSUB 100
     : GS = NM#
```

Columnar Orientation

One cannot read a trade magazine or paper today without encountering the phrase "user friendly." (So much of what is commonly termed "user friendly" turns out to be "user boring," but that's entirely another issue.) While we continue to emphasize the importance of writing programs which are user friendly, too many programmers continue to write programs (and input processors) which are a nightmare for those who inherit them. There is nothing friendly about:

```
2130 QQ$ = "N00000000000 00999999999
     Enter Gross Sales"
     : GOSUB 100 : GS = NM#
```

In this particular example, 'N' in Column 1 indicates that a numeric entry is required. Columns 2-12 and 14-24 indicate the minimum and maximum allowable values,

Figure 1

```
(1)  Access DRIVER.BA from the main menu.

(2)  Loading Input Processor...
     *Wait*

(3)  DRIVER.BA   LIST  .DO   TAXES .DO
     Q-SCRN.DO   Q-TEST.DO  ADRS  .DO
     BILLY .DO   STAT  .DO  HEXDEC.DO

     Enter Program Q-TEST
                   ------

(4)  Merging Q-TEST.DO
     Press [F4] to run when merge complete.
     *Wait*
     Ok

(5)  Press [F4].

(6)  Enter QQ$ Enter State,LEN=2,ALLUP
               ----------------------
     Enter State Illinois
                 --------
     <ERROR> Max Str Len 2
     Enter State il
                 --
     IL              2
     Enter State ca
                 --
     CA              2
     Enter State /
                 -

(7)  Enter QQ$ Enter Birthdate,DATE
               --------------------
     Enter Birthdate 11-8-56
                     -------
     11-08-56        561108
     Enter Birthdate 02/29/85
                     --------
     <ERROR> Date mm-dd-yy
     Enter Birthdate 02/28/85
                     --------
     02-28-85        850228
     Enter Birthdate /
                     -

(8)  Enter QQ$ Enter Age,MAX=70,MIN=16
               ----------------------
     Enter Age ten
               ---
     <ERROR> Num Reqd
     Enter Age 10
               --
     <ERROR> Min Value 16
     Enter Age 71
               --
     <ERROR> Max Value 70
     Enter Age 28
               --
     28                28
     Enter Age /
               -

(9)  Enter QQ$ /

     Ok
```

respectively. Columns 13 and 25 are used to indicate negative values. Column 26 through the end of QQ$ indicate the prompt. The meaning of the various columns will change depending on the contents of Column 1.

My criticism with this type of input processor stems from my general contempt for any column-oriented coding scheme. Column counting should have died with punched cards.

Multiple Variable

The first input processor I learned was of this type. Rather than passing a single string variable which is then parsed out, multiple variables are passed, each of which triggers some course of action within the subroutine. Consider:

```
2130 QQ$ = "Enter gross sales"
     : QL# = 0
     : QN# = 0
     : QM# = 9999999.99
     : GOSUB 100
     : GS = QS#
```

In this particular model:

QQ$ — indicates the prompt to be displayed on the screen.
QL# — is the maximum length of the string to be returned in QS$ or, if zero, indicates that a numeric value is to be returned in QS#.
QN# — is checked only if QL# is zero and indicates the minimum allowable value of QS#.
QM# — is checked only if QL# is zero and indicates the maximum allowable value of QS#.

This type of input processor is an improvement over the columnar orientation in that it is more self-documenting: At least you don't have to count columns. My criticism with this format is the number of variables that must be remembered and set.

A Better Input Processor

Q-SCRN: The input processor suggested here provides the following advantages over those previously mentioned.

1) All calls are to one routine only: GOSUB 100.
2) Only one variable is passed to the routine: QQ$.
3) Only two variables are used to return data; string variables are returned in QS$ and numeric values are returned in QS#.
4) Additionally, only QQ!, QQ#, QQ%, QS! and QS% are used by the subroutine. All other variables are available for main program use.
5) The key words contained within QQ$ are easy to learn and easy to code; no column counting.
6) The format of QQ$ is reasonably self-documenting.

Regarding comment number 6, consider the following. Even prior to a detailed explanation of the subroutine, the reader can probably determine the desired input for:

```
QQ$ = "Enter name,LEN=20"
QQ$ = "Enter Soc-Sec-Nbr,SSN"
QQ$ = "Enter hire date,DATE"
QQ$ = "Enter age,MIN=16,MAX=70"
QQ$ = "More?,YESNO"
```

The *Q-SCRN* coding rules: The general format for QQ$ is: QQ$ = "prompt,option-1,option-2, . . . ,option-n". Commas (,) are used as delimiters in QQ$. Everything up to the first comma is assumed to be the prompt and will be displayed for the user. Consequently, the prompt should not contain embedded commas. *Q-SCRN* uses the LINEINPUT verb, so if you want a question mark to be shown you must include it in the prompt. The various options, or key words, follow the prompt. Remember that commas are used to separate the options. Table 1 shows the *Q-SCRN* options.

The Source Code

The routine uses lines 100 through 199. There are three reasons for this. First, since BASIC must go to the top of the program and sequentially search for the beginning of a subroutine everytime a GOSUB is used, execution time can be saved by putting commonly used subroutines at the front of the program. Second, the range was limited to the one hundreds so other subroutines can be placed in 200 through 299, 300 through 399, etc. Third, using a three-digit line number saves 133 bytes over a four-digit line number and 233 bytes over a five-digit line number. This is of little concern to Model 2000 users, but is of some significance to Model 100 users. A more detailed explanation of some portions of the code follows.

Line 101 checks to see if the screen should be cleared prior to showing the prompt. The screen is cleared only once for a given prompt, so that any error messages will not be wiped out.

Line 102 displays the prompt, which is the contents of QQ$ up to the left-most comma.

Line 103 initializes the return values. QS$ is set to the null string and QS# is set to zero. This is necessary in order to allow a null entry (just pressing ENTER). Were these left out, QS$ and QS# would remain unchanged.

Line 105 checks to see if a slash (/) has been entered. In my programs a slash is allowed at all times and indicates that the user would like to quit. If you prefer to use some other character, such as the ESC key, or a character string, such as "QUIT", code that here. If you prefer to do that type of checking within the main body of your program, then delete Line 105.

Lines 106 through 114 determine which type of data is required and branches to the appropriate routine. If your QQ$ does not contain one of the options shown, then Line 119 will display an error message, show the value of QQ$ and STOP program execution.

Lines 120 through 125 process requests for string input. Line 120 is the first of several lines to use GOSUB 198. Lines 198 and 199 set QQ# to the amount specified after LEN=, MAX= and MIN=. For example, if QQ$="Enter name,LEN=20" then QQ# will be set to 20. Line 121 will print an error message if the string is longer than allowed. Line 122 makes sure some entry has been made, other than a null entry, unless the OPT option is used. Lines 123 through 125 will change all lowercase letters to uppercase if the ALLUP option is used.

Lines 130 through 146 process requests for numeric input. Line 131 makes sure some entry has been made, unless the OPT option is used, in which case default entry is zero. Lines 132 through 140 perform edits to make sure the entry is numeric: a plus sign or minus sign is allowed (but not required) as the first character, no more than one decimal point is allowed, and all other characters must

be zero through nine. Lines 142 through 146 check that the number is within the allowable range.

Lines 150 through 157 process the ACID option, or Add-Change-Inquiry-Delete. This option is useful for file maintenance or inquiry programs. As pointed out in Table 1, a variety of input is accepted but the returned values are standardized, simplifying the coding required by the main program. QS# is set to one, two, three or four, providing for use of the ON verb. For example:

Table 1

Keyword	Example	Effect	Returned Value
LEN	LEN=10	LEN indicates a string field is required, the length of which cannot exceed the value specified.	QS$ = string QS# = LEN(string)
MAX	MAX=80	MAX indicates a numeric field is required, the value of which cannot exceed the value specified.	QS# = value
MIN	MIN=-10	MIN is optional, but if used, must be used with MAX. The value entered by the user must be greater than or equal to the value specified. When MAX is used without MIN, the default minimum value is zero.	QS# = value
ACID	ACID	Allow entry of: ADD A add a CHANGE C change c CH ch INQUIRY I inquiry i INQ inq DELETE D delete d DEL del	QS$ = "A" QS# = 1 QS$ = "C" QS# = 2 QS$ = "I" QS# = 3 QS$ = "D" QS# = 4
ACK	ACK	ACKnowlege; display prompt then wait for user to press ENTER key.	n/a
OPT	OPT	OPTional; the user may just hit ENTER. However if an entry is made it must meet the conditions specified by the other options.	QS$ = "" (null) or depends on other options.
CLS	CLS	Clear the screen prior to displaying the prompt.	n/a
SSN	SSN	A valid social security number is to be entered in nnn-nn-nnnn format. (Hyphens required.)	QS$ = "nnn-nn-nnnn" QS# = nnnnnnnnn
DATE	DATE	A date is to be entered. Many formats will be accepted, but all are returned in mm-dd-yy format. For example: enter return 1-1-84 01-01-84 02/29/84 02-29-84 02/29/85 <ERROR>	QS$ = "mm-dd-yy" QS# = yymmdd

```
2130 QQ$ = "Enter option,ACID"
     : GOSUB 100
2140 ON QS# GOTO 2300, 2400, 2500, 2600
2150 CLOSE
     : END
```

Recall that QS# was initialized to zero, so if QS# does not equal one, two, three or four, then the user *must* have entered a slash (/) indicating he wanted to quit, hence Line 2150.

Lines 160 through 169 process requests for a Social Security number. I felt it was important to try to force the user to be careful in entering the number, so I require the entry to be in "nnn-nn-nnnn" format. This does not mean you must use it in that format; QS# is also returned with the numeric equivalent.

Lines 173 through 175 process requests for a "yes" or "no" response. As with the ACID option, a variety of inputs are accepted, but the returned values are standardized. QS$ will always be 'Y' or 'N' and QS# will always be '1' or '0.' I did not set QS# to '2' and '1', which would have made use of the ON verb possible as in the ACID option, because 'Y' and 'N' are commonly equated to '1' and '0' in Boolean algebra. If you prefer 2 and 1, change the QS# assignment in lines 173, 174, 177 and 179.

Lines 180 through 197 process DATE requests. Again, the dates are accepted in a variety of formats, but are always returned as "mm-dd-yy." The numeric equivalent of yymmdd is also returned in QS#. *Q-SCRN* includes a check for leap year. "02-29-84" would be valid but "02-29-85" would cause an error. January 6, 1939 could be entered as:

1/6/39	01/6/39	1/06/39	01/06/39
1-6-39	01-6-39	1-06-39	01-06-39

For all of the above, QS$ = "01-06-39" and QS# = 390106.

Making the Best of it with the Model 100

Q-SCRN can be used with the Model 2000 or the Model 100. Model 2000 users can, if desired, maintain a copy of *Q-SCRN* in each program that uses it, but Model 100 users don't have enough RAM to do that. I recommend the following, which will load *Q-SCRN* as needed.

1) Add the following (Line 10) to *Q-SCRN* and store the program as a text file (Q-SCRN.DO)

```
10 CLS
   : FILES
   : PRINT
   : QQ$ = "Enter program,LEN=6,ALLUP"
   : GOSUB 100
   : IF QS$ = "/" THEN MENU ELSE FL$ =
     QS$ + ".DO"
   : CLS
   : PRINT
   : PRINT "Merging "; FL$
   : PRINT "Press [F4] to run when merge
     complete."
   : MERGE FL$
```

2) All programs should begin with Line 10 indicating the first line of the program. I suggest GOTO 1000. This will branch around the subroutine and erase the Line 10 as in number 1. Programs should not include any lines

Full Featured, Proven ...

ACCOUNTING SOFTWARE

$15 PER SYSTEM

General Ledger
Accounts Receivable
Accounts Payable
Payroll *Including State & Local Taxes*
Inventory *With Purchase Order & Order Entry*

OR **ALL SYSTEMS ABOVE** FOR **$50** TOTAL

Runs on IBM (PC jr/PC/XT/AT), AT&T, EPSON, TANDY (1000/1200/2000), TI, WANG, COMPAQ, CORONA, SANYO, COLUMBIA*

Requires 128K RAM, 2 Floppies or Fixed Disk, 80 Column Printer, MS-DOS, PC-DOS 2.0 or later.

To do special FORECASTS, REPORTS, ANALYSES, etc., — You may use . . .

Lotus 1-2-3	Framework	Symphony
Dbase II/III	Wordstar	Condor
Multiplan	Word Perfect	My Word*

or any other Database, Word Processor or Spread Sheet package that you are now using with our data files.

The + (plus) key on the IBM keyboard functions as an ENTER key for data entry.

Exceptionally easy setup. From package to computing in less than *1 HOUR*.

All systems use generally accepted accounting principles.

Detailed manual is on diskette, you print it in your printer.

Disks are *NOT* Copy Protected!!

WORD PROCESSING

A WS-like word processor with Sort, Merge, Quick Print, More than 100 Operations, ASCII files

COMPLETE FOR $25

PRACTICING ACCOUNTANTS

Accountants Package, including: Client Write-Up with General Ledger, After-the-fact Payroll, Billing, and 1040 Tax Package, Everything you need for your Practice.

COMPLETE FOR $150

☆ ☆ ☆

ATTENTION DEALERS!!! Call or Write Now for the VERY BEST MARGINS (over 50%) in the Industry.

MCBS

MICRO COMPUTER BUSINESS SERVICES
12703 Stuebner Airline
Suite 202
Houston, TX 77014

TO ORDER CALL:
1-800-334-0854
Extension 826

For More Information
Call (713) 583-6828

Service Contract Available
Shipping Charges $5.00
Texas Residents add 6-1/8% tax
COD Charges $5.00
Payment by Check, MC or VISA
MONEY BACK GUARANTEE!

**Computer & software names are trademark or registered by their respective manufacturers.*

numbered 100 through 199. All programs should be stored as text files in order to use the MERGE verb.

3) Create DRIVER.BA as shown. DRIVER.BA is accessed from the main menu anytime a program is to be run:

```
10 CLS
   : PRINT
   : PRINT "Loading Input Processor . . ."
   : LOAD "Q-SCRN.DO",R
```

To familiarize yourself with the subroutine, create the following program, *Q-TEST*, and store as a text file.

```
10 QQ$ = "Enter QQ$,LEN=100"
   : GOSUB 100
   : IF QS$ = "/" THEN END ELSE QQ$ = QS$
20 GOSUB 100
   : IF QS$ = "/" THEN 10 ELSE PRINT QS$, QS#
   : GOTO 20
```

From the main menu, access DRIVER.BA. When prompted, merge Q-TEST.DO. Press F4 to run. *Q-TEST* will allow you to practice using *Q-SCRN*. *Q-TEST* will prompt for values of QQ$, then GOSUB 100 using the QQ$ you have specified. This will also allow you to check that *Q-SCRN* was keyed properly. A sample session is shown in Figure 1.

With a little practice you will feel very comfortable using *Q-SCRN*. New program development will be greatly simplified. Hope to see some of those programs here soon. Good luck!

Listing 1: DRIVER

```
10 CLS:PRINT:PRINT"Loading Input Process
or...":LOAD"Q-SCRN.DO",R
```

Listing 2: Q-SCRN

```
10 CLS:FILES:PRINT:QQ$="Enter Program,LE
N=6,ALLUP":GOSUB 100:IF QS$="/"THENMENUE
LSEFL$=QS$+".DO":CLS:PRINT:PRINT"Merging
 ";FL$:PRINT "Press [F4] to run when mer
ge complete.":MERGEFL$
100 'Screen Input Routine Version 2.0
101 IF INSTR(1,QQ$,",CLS")<>0 THEN CLS
102 PRINT LEFT$(QQ$,INSTR(1,QQ$,",")-1);
103 QS$="":QS#=0
104 LINEINPUT " "; QS$
105 IF QS$="/" THEN RETURN
106 IF INSTR(1,QQ$,",ACK")<>0 THEN RETUR
N
107 IF INSTR(1,QQ$,",LEN=")<>0 THEN 120
108 IF INSTR(1,QQ$,",MAX=")<>0 THEN 130
109 IF INSTR(1,QQ$,",ACID")<>0 THEN 150
110 IF INSTR(1,QQ$,",YESNO")<>0 THEN 173
111 IF INSTR(1,QQ$,",YES")<>0 THEN 176
112 IF INSTR(1,QQ$,",NO")<>0 THEN 178
113 IF INSTR(1,QQ$,",SSN")<>0 THEN 160
114 IF INSTR(1,QQ$,",DATE")<>0 THEN 180
119 PRINT "<ERROR> Bad QQ$=";QQ$:STOP
120 QQ%=INSTR(1,QQ$,",LEN=")+5:GOSUB 198
121 IF LEN(QS$)>QQ# THEN PRINT "<ERROR>
Max Str Len";QQ#:GOTO 102
122 QS#=LEN(QS$):IF QS#=0 THEN IF INSTR(
1,QQ$,",OPT")=0 THEN PRINT "<ERROR> Entr
y Reqd":GOTO 102 ELSE RETURN
123 IF INSTR(1,QQ$,",ALLUP")=0 THEN RETU
RN ELSE FOR QQ!=1 TO QS#
124  QQ%=ASC(MID$(QS$,QQ!,1)):IF QQ%=>97
 AND QQ%<=122 THEN MID$(QS$,QQ!,1)=CHR$(
QQ%-32)
125 NEXT QQ!:RETURN
130 'Nbr Reqd
131 IF LEN(QS$)=0 THEN IF INSTR(1,QQ$,",
OPT")=0 THEN 140 ELSE RETURN
132 QQ!=0 'Count decimal pts
133 FOR QQ% = 1 TO LEN(QS$)
134  IF QQ%=1 THEN IF MID$(QS$,QQ%,1)="+
" OR MID$(QS$,QQ%,1)="-"  THEN 139
136  IF MID$(QS$,QQ%,1)="." THEN QQ!=QQ!
+1:GOTO 139
137  IF MID$(QS$,QQ%,1)<"0" OR MID$(QS$,
QQ%,1)>"9" THEN 140
139 NEXT QQ%:IF QQ!<=1 THEN QS#=VAL(QS$)
:GOTO 142
140 PRINT "<ERROR> Num Reqd":GOTO 102
142 QQ%=INSTR(1,QQ$,",MIN")
143 IF QQ%=0 THEN QQ#=0 ELSE QQ%=QQ%+5:G
OSUB 198
144 IF QS#<QQ# THEN PRINT "<ERROR> Min V
alue";QQ#:GOTO 102
145 QQ%=INSTR(1,QQ$,",MAX=")+5:GOSUB 198
146 IF QS#>QQ# THEN PRINT "<ERROR> Max V
alue";QQ#:GOTO 102:ELSE RETURN
150 'A-C-I-or-D Reqd
151 IF QS$="ADD" OR QS$="add" OR QS$="A"
 OR QS$="a" THEN QS$="A":QS#=1:RETURN
153 IF QS$="CHANGE" OR QS$="change" OR Q
S$="C" OR QS$="c" OR QS$="CH" OR QS$="ch
" THEN QS$="C":QS#=2:RETURN
155 IF QS$="INQUIRY" OR QS$="inquiry" OR
 QS$="I" OR QS$="i" OR QS$="INQ" OR QS$=
"inq" THEN QS$="I":QS#=3:RETURN
157 IF QS$="DELETE" OR QS$="delete" OR Q
S$="D" OR QS$="d" OR QS$="DEL" OR QS$="d
el" THEN QS$="D":QS#=4:RETURN
159 PRINT "<ERROR> Add, Change, Inquiry
or Delete":GOTO 102
160 'SSN Reqd
161 IF LEN(QS$)=0 THEN IF INSTR(1,QQ$,",
OPT")<>0 THEN RETURN ELSE PRINT "<ERROR>
 SSN Reqd":GOTO 102
162 IF LEN(QS$)<>11 THEN 169
163 FOR QQ% = 1 TO 11
```

```
164  IF QQ%=4 OR QQ%=7 THEN IF MID$(QS$,
QQ%,1)="-" THEN 166 ELSE 169
165  IF MID$(QS$,QQ%,1)<"0" OR MID$(QS$,
QQ%,1)>"9" THEN 169
166 NEXT QQ%
167 QS#=VAL(MID$(QS$,1,3) + MID$(QS$,5,2
) + MID$(QS$,8,4)):RETURN
169 PRINT "<ERROR> SSN Fmt xxx-xx-xxxx":
GOTO 102
173 IF QS$="YES" OR QS$="yes" OR QS$="Y"
 OR QS$="y" THEN QS$="Y":QS#=1:RETURN
174 IF QS$="NO" OR QS$="no" OR QS$="N" O
R QS$="n" THEN QS$="N":QS#=0:RETURN
175 PRINT "<ERROR> YES or NO":GOTO 102
176  'YES or NO Reqd Default YES
177 IF LEN(QS$)=0 THEN QS$="Y":QS#=1:RET
URN ELSE GOTO 173
178  'YES or NO Reqd Default NO
179 IF LEN(QS$)=0 THEN QS$="N":QS#=0:RET
URN:ELSE GOTO 173
180  'Date mm-dd-yy Reqd
181  QQ%=INSTR(1,QS$,"/"):IF QQ%<>0 THEN
 MID$(QS$,QQ%,1)="-":GOTO 181
182 IF MID$(QS$,2,1)="-" THEN QS$="0"+QS
$  'Fix m-dd-yy
183 IF MID$(QS$,5,1)="-" THEN QS$=LEFT$(
QS$,3)+"0"+RIGHT$(QS$,LEN(QS$)-3)  'Fix m
m-d-yy
184 IF LEN(QS$)<>8 THEN 191 ELSE FOR QQ!
 = 1 TO 8
185  IF QQ!=3 OR QQ!=6 THEN IF MID$(QS$,
QQ!,1)<>"-" THEN 191 ELSE 187
186  IF MID$(QS$,QQ!,1)<"0" OR MID$(QS$,
QQ!,1)>"9" THEN 191
187 NEXT QQ!
188 QQ%=VAL(MID$(QS$,4,2)):IF QQ%<1 THEN
 191
190 ON VAL(LEFT$(QS$,2)) GOTO 192,194,19
2,193,192,193,192,192,193,192,193,192
191 PRINT "<ERROR> Date mm-dd-yy":GOTO 1
02
192 QQ!=3:GOTO 195
193 QQ!=2:GOTO 195
194 QQ!=VAL(RIGHT$(QS$,2))/4:QQ!=ABS(QQ!
=INT(QQ!))
195 QQ!=QQ!+28:IF QQ%>QQ! THEN 191
197 QS# = VAL(MID$(QS$,7,2) + MID$(QS$,1
,2) + MID$(QS$,4,2)):RETURN
198 QQ!=INSTR(QQ%,QQ$,","):IF QQ!=0 THEN
 QQ!=LEN(QQ$)+1
199 QQ!=QQ!-QQ%:QQ#=VAL(MID$(QQ$,QQ%,QQ!
)):RETURN
```

Listing 3: Q-TEST

```
 10 QQ$="Enter QQ$,LEN=100":GOSUB 100:IF
 QS$="/" THEN END ELSE QQ$=QS$
 20 GOSUB 100:IF QS$="/" THEN 10 ELSE
 PRINT QS$,QS#:GOTO 20
```

PCM

Call Delker. The accessible experts.

TANDY Radio Shack ADVANCED TECHNOLOGY SERIES AND AFFORDABLE TOO ™

Tandy 1000 represents the state of the art in performance, quality and price breakthrough.

When you buy from Delker Electronics you can be sure you are getting high quality factory direct equipment at very competitive prices. Plus, you are getting the experience in computer maintenance and sales that comes from many years of experience in the field.

DELKER ELECTRONICS

DELKER ELECTRONICS, INC.
P.O. Box 897 408 C Nissan Blvd.
Smyrna, TN 37167

800-251-5008 615-459-2636 (Tennessee)
800-545-2502 (Tennessee) 615-254-0088 (Nashville)

Visa, Mastercard and American Express accepted.

It's the Holmes Engineering/PCSG "chipmunk"

PORTABLE Disk Drive

for the Model 100/200

$599
Includes a library of six powerful programs

Uses the main menu concept. You see the disk directory instantly, arranged on your M-100 screen like your main menu. Just move the widebar cursor and transfer files with a function key. You can run a file directly from the diskette with the ENTER key. Uses 3½" microfloppy diskettes that have a rigid plastic casing and a metal core. They're tough and nonflexible. You can carry several in a shirt pocket without damage. There's 358K on a diskette. Ten of these in your briefcase and you've got 3½ megabytes.

Drive weighs only three lbs. and it works directly from the 110 outlet and recharges at the same time. It recharges in six hours with thousands of pages transferred between charges. It's compact, with dimensions of 2¼" x 5½" x 7.5"; and fits easily into your briefcase along with your Model 100 or 200.

Machine code programs, BASIC programs, *Lucid* files and documents all are saved and retrieved with no protocol–instantly, ready to run.

In a special association, Holmes Engineering and PCSG have worked together combining the hardware knowledge of Holmes and the software expertise of PCSG. The result is a product that can only be regarded as excellent.

You see the disk directory instantly; works just like the main menu

Here is what is really exciting. The portable disk drive has Random Access. Included as part of the operating system in the drive (ROM) is a very powerful disk BASIC.

This means that you can have BASIC programs that will access the diskette and read and write records directly on the diskette.

Just imagine yourself with this kind of capability.

Database–The portable disk drive stores your mailing list, inventory items, part #s and descriptions or any other data that you need to recall.

358K on a diskette

Invoice (purchase order)–At the touch of a button you can print out your sequentially numbered, professionally done invoices. This is truly professional invoicing capability.

Purchase orders are just as easy.

Sort – This excellent utility allows you rapid sorting of any records you have compiled. You can write the newly sorted list back in the same file on the diskette or to a new file.

Telecom interface – If you are a user that likes to access other computers or databases (for example CompuServe) by telephone then this powerful facility alone is worth the price of the disk drive. You can automatically download and upload information onto the diskette.

Calendar – Everyone who has seen this program has said, "This is the first calendar/diary/scheduler on any computer anywhere that I can use. It is so functional."

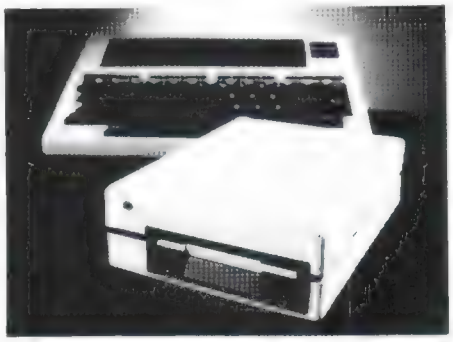
The calendar program is usable for two reasons, first it is designed correctly, and second you have the memory (358K) on the diskette to log and access a tremendous amount of notes over a long period of time.

Personal Finance Manager – This wonderful program truly lets you keep track of your finances.

All your records are kept on the diskette. Bank accounts (checking and savings) and charge accounts such as MasterCard and Visa.

We at PCSG believe we have the ultimate Model 100 system, the Portable Disk Drive plus the *Lucid* spreadsheet on snap-in ROM, *Write ROM* word processing and the new 64K RAM expansion now available from PCSG.

We want you to find out for yourself at no risk. If you aren't totally satisfied within 30 days, simply return the disk drive for a full refund. Priced at $599.95, including the software library. MasterCard, Visa, COD.



PORTABLE
COMPUTER
SUPPORT
GROUP

1-214-351-0564

11035 Harry Hines Blvd., #207
Dallas, Texas 75229

PCSG offers a 30-day/money-back trial!

Manipulating RAM Files in the Tandy 200

By Carl Oppedahl

After downloading a document file from my TRS-80 Model I to my Tandy 200, there was my .DO file on the screen. Square root signs appeared where tabs should have been, and carriage returns did not display correctly.

I figured out an easy way to fix the .DO file and it is described here.

If you read this article you will learn how .DO files are stored in RAM in the Tandy 200, and you'll see how some ROM subroutines may be used in BASIC. You can use these techniques in many other ways. The methods I describe here also apply to the Model 100.

The Problem

Because of the limited memory capacity of the Tandy 200 I sometimes load a file from the 200 to a larger machine — in my case a TRS-80 Model I. Later I load the file back to the Tandy 200. Usually I find the document file to be in the same condition as before the transfer, but on several occasions the file looked strange. A square root sign appeared where a tab should have been, while a plus/minus sign showed up where the carriage return should have been. This is illustrated in Listing 1.

A review of my actions while using the Model I revealed the cause. While the document file was on a Model I disk, I had edited the file using *Scripsit*. *Scripsit* reads a file from disk into RAM, then when editing is finished it writes the file back out onto disk.

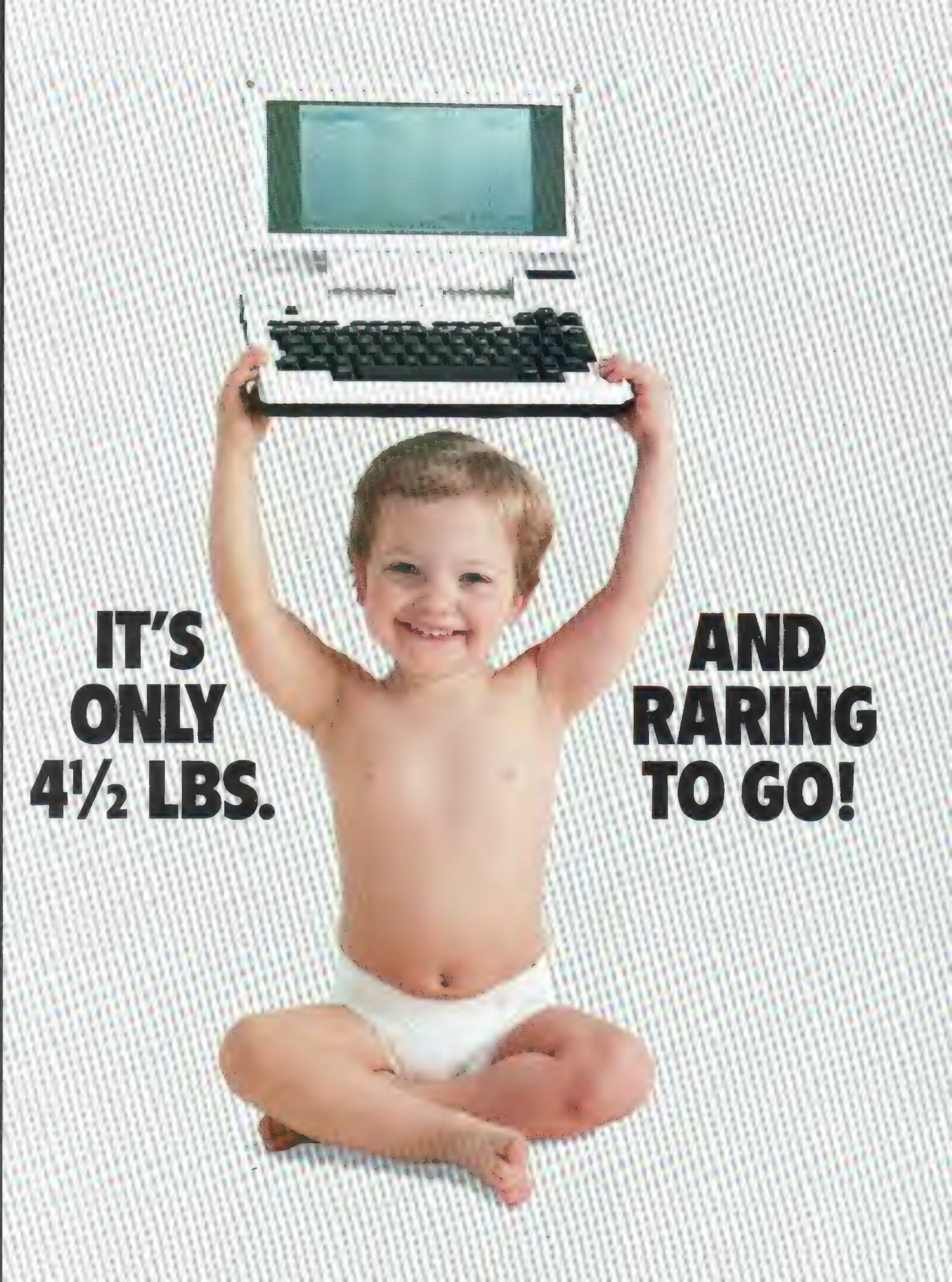
As I later learned, the reason the .DO file in the Tandy 200 looked funny is it came from a Model I disk file that had been written out by *Scripsit*. *Scripsit* had modified the text before writing it back to disk. *Scripsit* stores control characters such as the tab and carriage return with bit 7 on. The tab, stored by the Tandy 200's *TEXT* program as a decimal value of nine, is stored by *Scripsit* as a 137 (9+128). Similarly, the carriage return, stored by *TEXT* as a decimal value of 13, is stored by *Scripsit* as a 141 (13+128). What's worse, *TEXT* indicates end-of-paragraph with a two-character sequence, `CHR$(13)+CHR$(10)`, or with a carriage return and line feed, while *Scripsit* uses a one-character sequence, `CHR$(141)`.

The character with the value of 137 shows up on the Tandy 200 screen as a square root sign, which explains the spurious square roots of Figure 1. The 141 character shows up as the plus/minus character, which explains the plus/minus signs of Figure 1.

Designing a Solution

The problem, then, was that my .DO file had strange, inappropriate characters in it. The simplest solution would have been to go through the document with *TEXT* searching for the incorrect characters with F1 (Find), deleting

(Carl Oppedahl is a lawyer specializing in technological litigation. He is the author of Inside the TRS-80 Model 100, *published by Weber Systems, Chesterland, Ohio.)*


IT'S ONLY 4½ LBS.
AND RARING TO GO!

The Tandy 200 is the one portable for all.

Meet the New Generation of Portable Computing

Our celebrated Model 100 set a new standard in portable computing. Now we've done it again! Introducing the Tandy 200, another true breakthrough, featuring advanced features *you* requested. You get more built-in software, a bigger screen and a larger memory. All this in a system that measures just $2^1/_4 \times 11^3/_4 \times 8^1/_2$", and goes wherever you go because it's completely battery powered.

Six Built-In Programs Including Multiplan™

For complex spreadsheet analysis and calculations, we put popular Multiplan software into the Tandy 200's permanent memory. It's easy to do sales forecasts, profit and loss projections, budgeting, pricing, engineering calculations and more.

Comes with Five More Powerful Programs

An improved version of the Model 100's easy-to-use word processing program makes the Tandy 200 especially useful for journalists, salespeople, students and anyone who needs to write letter-perfect memos, reports and correspondence in a hurry. Edit, delete and move blocks of text with the touch of a convenient function key.

Four other "instant-on" programs let you use the Tandy 200 as your personal appointment calendar, address and phone directory and telephone auto-dialer/directory (the Tandy 200 generates tone dialing pulses, so you can use it with long-distance services). A much more powerful built-in program for communications makes it easy to access other computers by phone, as well as national information networks. Resident BASIC language lets you write your own programs, too.

Why 40 Columns Are "Bigger" than 80

Take a look at the 80-column screens on other portables, and you'll see why we chose a 16 × 40-character format. Characters on 80-column displays are tiny and difficult to read. Tandy 200's flip-up liquid crystal display has 240 × 128 resolution for big, clear graphics and easy-to-read characters. Tandy 200 is the perfect take-along tool for word processing and spreadsheets—without eyestrain.

We also increased the standard memory size to 24,000 characters, expandable to 72K.

The Best in Technology for Under $1000

Whether you're a student, scientist, busy professional or home computer user, the Tandy 200 is a smart investment. The Tandy 200 even has parallel printer, RS-232C, cassette and bar code reader interfaces for added versatility. You can even add disk storage and a monitor for a complete desktop computer system.

Adopt One Today!

Step up to the powerful software and impressive capabilities of the Tandy 200 portable computer for just $999 (26-3860). Best of all, the Advanced Technology Tandy 200 represents the state of the art in performance, quality and price breakthrough (because we've introduced the latest technology for over 60 years). Stop by your local Radio Shack Computer Center, or participating Radio Shack store or dealer and "size it up" today!

Prices apply at Radio Shack Computer Centers and at participating Radio Shack stores and dealers. Multiplan/TM Microsoft Corp.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Send Me a 1985 Computer Catalog!
Mail To: Radio Shack, Dept. 86-A-224, 300 One Tandy Center, Fort Worth, Texas 76102

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
PHONE _____

them, and inserting the correct characters. But my .DO file was very big, and the editing would have taken many hours. So I resolved to write a program to fix everything automatically.

A False Start

Usually when a file must be massaged in some way, the easiest way to get the job done is with a BASIC program. Open the .DO file for input, bring in the characters one by one and print them out (with changes or substitutions as necessary) to a second .DO file. By the time the program has finished, there are two .DO files in RAM of roughly equal size. I could not do this because the existing .DO file filled almost all of my RAM area. There was not enough room for the output file.

One way to proceed would have been to write the .DO file out onto cassette tape, then kill the RAM .DO file, then let the BASIC program open the cassette file for input, sending its output to RAM. But cassette storage is slow and sometimes unreliable, and I had no recorder nearby.

So I resolved to modify the .DO file directly where it sat in RAM. The advantage was that no cassette recorder was required, and the file corrections would be completed quickly. The disadvantage was that I had to figure out how to do it (using PEEKs and POKEs) without damaging my RAM files and possibly bringing about the dreaded return to January 1, 1900. The point of this article is to describe the program I came up with.

The general strategy was this:

- — Locate the beginning of the .DO file in RAM
- — Find an aberrant character (square root or plus/minus sign) by PEEKing to successive memory locations
- — POKE in the correct character (tab or carriage return
- — Stop when the end of the file is reached.

I will discuss these steps one by one.

Locating the .DO File in RAM

Many readers are familiar with the file directory used in the Tandy 200, and it is similar to that used in the Model 100 (see my article "The Searcher," *80 Micro*, June 1984, Page 157). Briefly, the directory resides at F2B5 through F4AF Hex in the protected system RAM area. Each group of 11 bytes represents one file — the first byte signals the file type, the second and third bytes give the file's starting address, and the remaining bytes contain the filename.

One way a BASIC program could get the starting address of a file, then, is by examining one by one the entries in the directory. If the file type is 192, in other words a non-killed .DO file, then the filename should be checked. If the filename matches, then the address is given in the second and third bytes of the entry.

This is rather a tedious kind of searching, though it can be done well enough in BASIC by means of the PEEK command.

Another way to get the starting address of a .DO file is by means of certain ROM subroutines (CHKDC at 6E4D and GTXTTB at 6E86). These routines cannot usefully be called directly from BASIC, however, because they return their information in registers of the CPU, and the CALL command in BASIC makes no provision for returning register values to BASIC.

To get around the limitation in the CALL command, a hardcore machine language programmer would use BASIC to call a machine language program which would itself call CHKDC and GTXTTB, take the values from the CPU registers and POKE them into a BASIC variable or put them in a place where they could be PEEKed by BASIC. Such a strategy requires resetting HIMEM, POKEing machine language values into RAM, CALLing them and messing up any other machine language programs above HIMEM. This would make the program very complicated.

The Clean Way to Find the .DO File

I finally worked out a way to get the starting address of the .DO file without resorting to machine language — what is needed are PEEKs and the VARPTR function. Recall that VARPTR is described in the Tandy 200 owner's manual as applying to numerical variables. As it happens, VARPTR may be applied to string variables (explained in "The Searcher" article) and may also be applied to open files.

Figure 1. Spurious Characters

Here is a line of text. ± Here is another line. ± Here is a tab √ character.

When a file has been opened, a file number is always assigned, say file number 1. The expression VARPTR(#1) then yields the address of what is called the file control block (FCB), located in the BASIC working space below HIMEM. (The number of FCBs in your Tandy 200 at a particular moment is determined by the MAXFILES command.)

The FCB contains various pieces of information, depending on the file type — CAS:, RAM: and so on. In the case of the FCB for a RAM file, the third and fourth bytes of the FCB give the address of the directory entry for the RAM file.

For example, assume a RAM .DO file has been opened as file 1. After execution of I=VARPTER(#1), then I points to the FCB for the file. In other words, PEEK(I) reveals the contents of the first byte of the FCB, PEEK(I+1) reveals the contents of the second byte, and so on. Here what we want is to extract from the FCB the location of the directory entry for the file; the information is in the third and fourth bytes. After execution of J=PEEK(I+2)+256*PEEK(I+3), then J is the address of the directory entry. (Strictly, J points to the second byte of the directory entry.

Given J, it is now an easy matter to extract from the directory entry the address where the .DO file begins in RAM. Here is how it is done: BEG=PEEK(J)+256*PEEK(J+1) assigns to BEG the address where the .DO file begins.

Listing 1: *Scripsit* Correction Program

```
10 INPUT "filename";I$:OPEN I$ FOR INPUT
 AS 1:I=VARPTR(#1):CLOSE:J=PEEK(I+2)+256
*PEEK(I+3):BEG=PEEK(J)+256*PEEK(J+1)
20 IF PEEK(BEG)=26 THEN PRINT"Done":BEEP
:END:ELSE IF PEEK(BEG)=141THEN POKE BEG,
13:BEG=BEG+1:CALL 27489,10,BEG ELSE IF P
EEK(BEG)=137 THENPOKE BEG,9
40 BEG=BEG+1:GOTO 20
```

Finding Aberrant Characters

Once the BASIC program has figured out where to find the .DO file, it is easy to locate the square root signs and plus/minus signs. This is done by PEEKing to successive memory locations, looking for the value 137 or 141.

When an aberrant character has been found, it is necessary to POKE in the correct value. This is the first truly dangerous part of the program. If an error occurred in typing in the program, the program may run amok, POKEing numbers into parts of memory other than the .DO file. The likely result is a cold start — a return to the dreaded January 1, 1900. Be very careful with all POKEs. If you are not sure you typed in the program right, make backups of all your files before testing the program.

Anyway, to fix an incorrect tab the correct code is `IF PEEK(BEG)=137 THEN POKE BEG,9`.

Correcting an incorrect carriage return is somewhat more difficult. When the *Scripsit*-style carriage return is found, it is easy enough to POKE in the value 13, but a line feed must also be inserted, moving up everything above it in the .DO file.

Fortunately, the ROM subroutine INSCHR (at 829C Hex or 33436 decimal), which inserts a character to a .DO file, may be used from BASIC. (In the Model 100 it is at 6B61 Hex or 27489 decimal.) This routine "bumps" everything in the user RAM area up one position to make room for the inserted character. The move changes the boundary between .DO and .CO files and the extreme boundary of the .CO files, and INSCHR updates the pointers in system RAM that keep track of these boundaries. For example, the number of free bytes that would appear at the main menu diminishes by one each time INSCHR is called.

Here is how the *Scripsit* carriage return may be corrected: `IF PEEK(BEG)=141 THEN POKE BEG,13:BEG=BEG+1:CALL 33436,10,BEG`.

Stopping at End-of-File

It is of extreme importance that the BASIC program stop its POKEing activity before wandering past the end of the .DO file. If the POKEing were to continue, it would mess up the system RAM area and would certainly cause a return to January 1, 1900. Since .DO files end with a Control-Z, or ASCII 26 decimal, the right thing to do is watch for a 26, like this: `IF PEEK(BEG)=26 THEN PRINT"DONE":BEEP:END`.

Out of Memory

This program increases the size of the .DO file for each carriage return encountered. This goes hand in hand with the fact that an otherwise identical *Scripsit* .DO file and TEXT.DO file will have different file sizes (the *Scripsit* .DO file is smaller).

There is the possibility that if the .DO file comes close to filling up memory, the BASIC program may try to cause the .DO file to exceed the available memory. The BASIC program does not protect for this possibility explicitly, but the subroutine INSCHR will simply return (with the carry flag set, though BASIC won't know that) without doing any further damage in the event that memory exceeded.

Conclusion

Through this article I have shown you a little about .DO file structure, and have illustrated some techniques for getting back and forth between machine language and BASIC. PCM

COCO-UTIL

Only $24.95

"The Connecting Link"

CoCo-Util is a valuable utility program that allows you to transfer Tandy Color Computer disc files to your MS-DOS machine. You may also transfer MS-DOS files to a Color Computer disc. CoCo-Util will save you countless hours of retyping. . .a great new utility.

NEED MORE MEMORY?

Set of 9 chips, 64K Upgrades. Great Bargain

Only $14.95

MS-DOS ADVENTURES

Our popular Hi-Res Color graphic Adventures. . .Calixto Island, Black Sanctum, Shenanigans, Sea Search, Trekboer are now available for MS-DOS computers to challenge and entertain you.

Only $29.95 per disc

PRINTERS

New Star Gemini SG-10 Model printer. 120 cps w/true descenders, proportional spacing. New model for 1985.

star

Our Great Price, Only $249.95

QUALITY VIDEO MONITORS

We carry both monochrome and color monitors at the best prices—several brands available. Call for more details.

Priced from $109.95

TILT SWIVEL BASE

Now you can adjust your monitor for best viewing angle. Left, right, up, down. Use with any monitor. Compare with units that sell for much more. Our great price—Only $14.95

DISC STORAGE CASE

Attractive, heavy duty, smoked acrylic case. Holds and protects up to 75 5¼" discs with six moveable, indexed dividers.

SUPER BARGAIN ONLY $9.95

storage box along with ten discs for only $21.95.

10 5¼' discs in attractive styrene box—Only $12.95

SHIPPING: All orders under $100 please add $2 regular, $5 air. All orders over $100 please add 3% regular, 8% air. California residents please add 6% sales tax. Orders outside the continental U.S. check with us for shipping amount, please remit U.S. funds. Software authors—contact us for exciting program marketing details. We accept MasterCard and VISA. Distributed in Canada by Kelly Software

FREE—Send for Our Latest Product Flyer!

MD Mark Data Products

24001 ALICIA PKWY., NO. 207 MISSION VIEJO, CA 92691 (714) 768-1551

Factory Authorized

Radio Shack® DEALER

Discover a Whole New World of TV Entertainment!

SATELLITE SYSTEM

User Installable! Over 100 Channels! Uncut Movies! Specials! Sports! News! Weather! Music Videos!

U.L. Approved

85° LNA

Entire System **$1999**

- Totally Automatic System With 10½-Ft. Dish, Receiver and Wireless Remote Control
- Comes With All Accessories Needed for Complete Do-it-Yourself Installation
- Descrambler Ready

Handy Wireless Remote Lets You Change Channels, Control Volume and Fine-Tune

Now you can enjoy it all! 24-hour non-stop entertainment that you simply can't get from local TV, cable, and even video tape! You'll see the hottest stars, recent uncut and uninterrupted movies, the ultimate in sports coverage—and the best part is, there are no monthly fees to pay. You'll see exciting family-oriented programs, plus music videos, FM broadcasts and movies with provision for stereo, and 24-hour news coverage on both TV and radio. It all adds up to out-of-this-world TV viewing at a down-to-earth price. Come in today and discover how easy it is to enjoy satellite programming! Includes complete installation manual.

(817) 825-4027

NOCONA ELECTRONICS • Box 593 • Nocona, TX 76255

UPS Shipping

A RADIO SHACK DEALER

The Device Driver:
An MS-DOS Power Feature

By Danny Humphress
PCM Technical Editor

Over the past 15 months, "MS-DOSsier" has covered a lot of ground. As we move further into our second year of this column, we are evolving from a beginner's column to one for the MS-DOS power user — the person who feels comfortable with MS-DOS, but has not yet mastered all the power available to him.

This month, we begin covering one of those "power" features — device drivers. While powerful in function, device drivers are surprisingly easy to take advantage of.

Device Drivers

As we have talked about in earlier issues, a "device" is simply a piece of hardware attached to the computer. Once you've physically connected a device, you must logically attach it to MS-DOS. That is, you have to tell MS-DOS the device is there and supply it with the interface software. This interface software is called a "device driver."

MS-DOS has built-in standard device drivers for common devices such as the video display, keyboard and disk drives, so there is no need for us to supply one. Other devices such as mouse boards and network boards usually require a software driver.

Software drivers are generally supplied when you purchase the device. The ViaNet board, for example, includes a device driver called NETWORK.SYS to drive the network card. The DigiMouse/Clock Controller Board comes with MOUSE.SYS and CLOCK.SYS device drivers.

Device drivers are not just for add-on hardware devices. Some drivers are used to change the way a common device works. ANSI.SYS, for example, is used to make the video screen and keyboard behave like a standard ANSI terminal (American National Standards Institute). Others, like LPDRVR.SYS on the Tandy 1000, change the way the printer reacts to control codes. A device driver is available through Tandy's Express Order Software that allows you to use part of your computer's RAM as a disk drive.

Installing a Device Driver

In order for your computer to recognize and use a particular device driver, you need do two things. First, you must copy the device driver file to the root directory of your MS-DOS boot disk or hard disk. Then, MS-DOS must be told the device is a part of its "configuration" when the system boots. This is done via the CONFIG.SYS file.

While the filename CONFIG.SYS would lead you to believe it is a device driver, it is actually a file that lists the device drivers to be included in the system when it is booted. It is also used to set certain system parameters (more on that later).

When you turn on or reset your computer, the computer looks for a CONFIG.SYS file on the boot disk for

information about the system configuration. If there is no CONFIG.SYS on the disk, MS-DOS assumes a "standard" configuration, then goes on to the AUTOEXEC.BAT file (if one exists) or to the normal date and time questions.

If MS-DOS finds a CONFIG.SYS file, it uses it to set system parameters and to find the names of the device drivers it needs to load. If, for example, you used CONFIG.SYS to tell DOS that you're using the ANSI.SYS console driver, it would look for the ANSI.SYS file and load it into memory as part of DOS.

The CONFIG.SYS file may be created with the EDLIN line editor, with a COPY CON: TO command, or with just about any text editor. (For more information on EDLIN, see the "MS-DOSsier" columns in the August 1984 through November 1984 issues of PCM.)

The lines in a CONFIG.SYS file loosely resemble a programming language. Like a language, MS-DOS recognizes certain key words in the CONFIG.SYS file as having specific meanings. Briefly, the keywords recognized are: BUFFERS, FILES, BREAK, SHELL and DEVICE.

The DEVICE command is used to tell MS-DOS to load a certain device driver when MS-DOS is booted (started up). If we wanted the ANSI.SYS and MOUSE.SYS drivers to be loaded when we booted, we would need these two commands in the CONFIG.SYS file:

```
DEVICE = ANSI.SYS
DEVICE = MOUSE.SYS
```

The first thing MS-DOS will do after loading is to load ANSI.SYS and MOUSE.SYS into memory and attach it to DOS. These device drivers will be in effect until the system is re-booted or turned off.

It is important to remember that the commands in the CONFIG.SYS file are not in effect until you reset your computer. That is the only time MS-DOS looks at CONFIG.SYS and configures the system.

ANSI.SYS

By designing their programs to work with ANSI terminal standard commands, programmers can easily move their applications from one machine to another without having to worry about video display and keyboard differences.

The ANSI.SYS device driver, which comes with all of Tandy's MS-DOS machines as well as most other compatibles, makes your display and keyboard behave according to the ANSI standards. While the keyboard and display accept the ANSI control codes, their normal operation is not affected. That is, programs that are not designed to work with ANSI.SYS will not be affected by its presence.

Many programs require that you have the ANSI.SYS driver installed. Among those available from Tandy that require ANSI.SYS are *Videotex Plus* and *filePro 16*. To use them, you must make sure this line is part of your CONFIG.SYS file:

```
DEVICE = ANSI.SYS
```

Even if you aren't a programmer, ANSI.SYS allows you to do some really nice things such as changing the screen colors and defining function keys. Details on this next month.

We'll also discuss some of the other available device drivers next month. Some of these drivers, such as the KEYCNVRT.SYS keyboard driver and the LPDRVR.SYS printer driver are specific to the Tandy 1000. We'll also take a look at DUMPCGP.SYS and DUMPBW.SYS, the graphics screen dump utilities. PCM

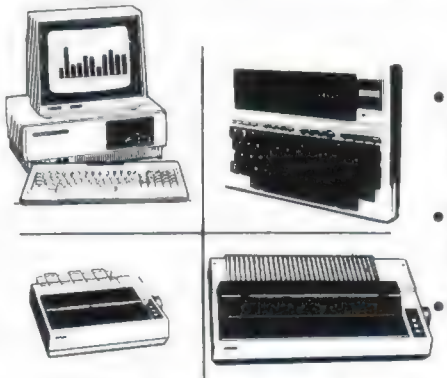
MARYMAC INDUSTRIES INC.

800-231-3680

Radio Shack TRS-80's

EPSON PRINTERS

People you Trust to give you the very best!



- Lowest Discount Prices
- Reliable Service
- Quality Products

"World's largest independent authorized TANDY Dealer"

22511 Katy Fwy., Katy (Houston) Texas 77450

(713) 392-0747 Telex 774132

TANDY 1000
INTERNAL 10 MEG HARD DRIVE $549.

Boots directly from Hard Drive using Standard Model 1000 DOS and looks like drive C. Price includes Controller, Hard Drive, Cables, Software, and Instructions. Installation only requires a screwdriver. External units add only $100.

Southwestern Digital

1-713-480-3296

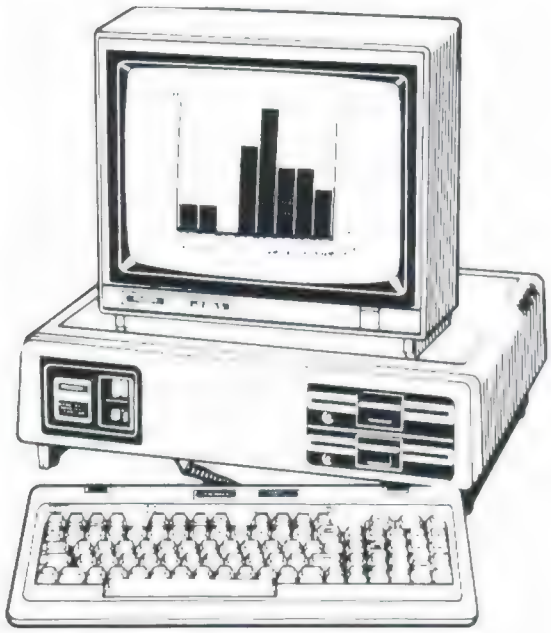
2515 W. Main #337 Houston, Texas 77098

Terms: All prices listed are pre-discounted for cash. Visa, Mastercard add 5% charge fee. COD add $2.. UPS ground service add $10. UPS air service add $20 All Texas sales add 6.125% state sales tax

NOCONA

ELECTRONICS

TRS-80 Model 2000 by Tandy*



MS-DOS Based

Computer System

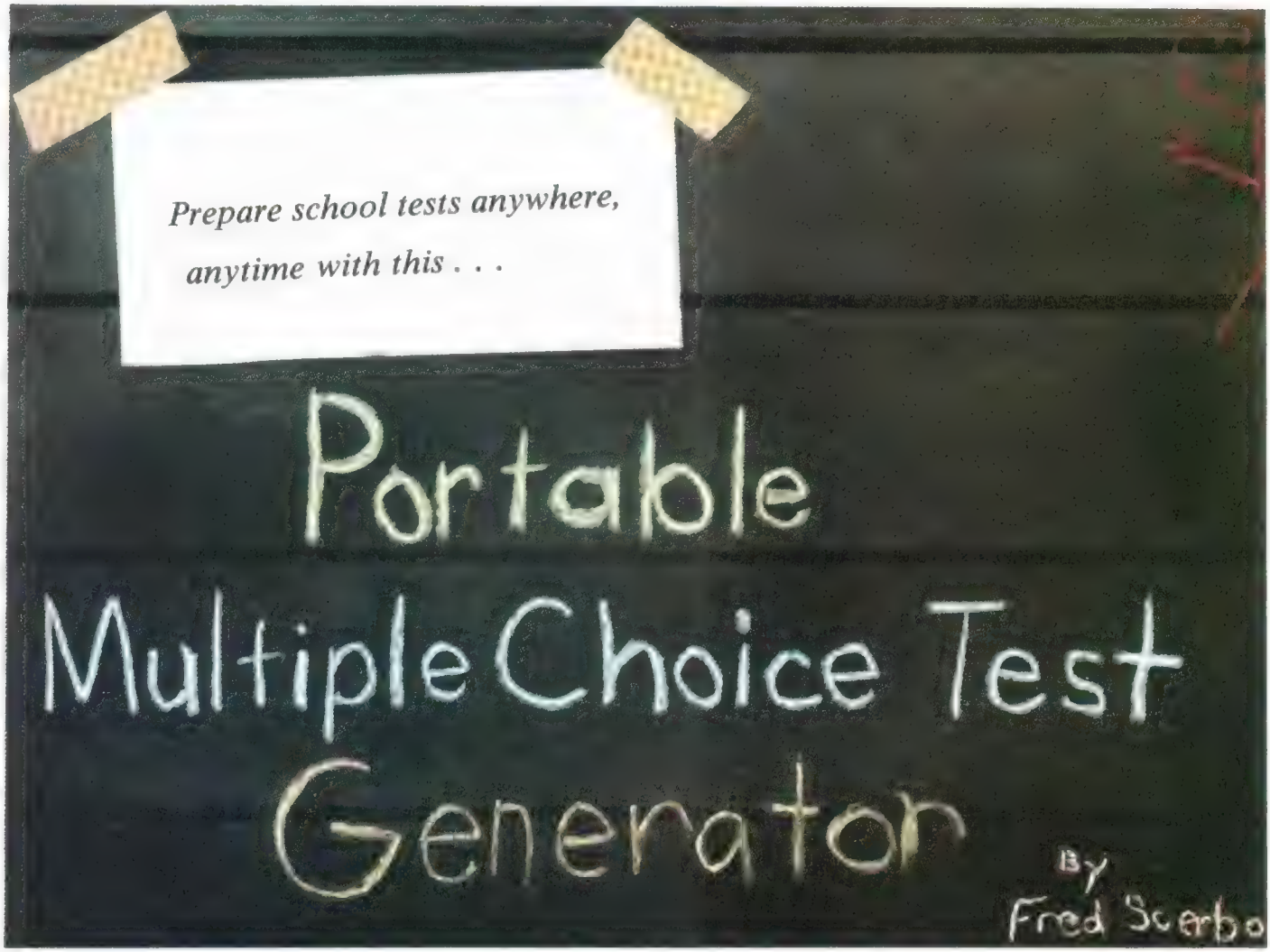
256K MOD2000 W/MONO MONITOR	1319.00
256K MOD2000 W/COLOR MONITOR	1649.00
256K MOD 1200 HD 2 DR.	1499.00
128K MOD 1000 W/MONO 1 DR.	839.00
128K MOD 1000 W/COLOR 2 DR.	1199.00
256K MOD 1000 W/HARD DISK	1439.00
24K MOD 100 PORT	399.00
24K MOD 200 PORT	769.00
64K MOD 4 2 DR.	845.00
512K MOD 6000 2DR	3099.00
80K MOD 12 2 DR.	1199.00
15 MEG HARD DISK	1329.00
35 MEG HARD DISK	2059.00
DMP 105 PRINTER	145.00
DMP 110 PRINTER	269.00
DMP 120 PRINTER	249.00
DMP 430 PRINTER	585.00
DMP 2100P PRINTER	1019.00
CGP 220 COLOR PRINTER	499.00
DWP 510 PRINTER	999.00
TRACTOR DWP 210	99.00
TRACTOR DWP II/510	159.00
TRACTOR FEED DMP2100P	99.00
PARALLEL PRINTER SWITCH	96.00
PARALLEL PRINTER CONTROLLER	179.00
PARALLEL PRINTER CABLE	29.00
DISK VIDEO INTERFACE	399.00

TRS-80 Model 100 by Radio Shack

100% RS COMPONENTS NO FOREIGN DRIVES OR MEMORY FULL WARRANTY
ALL RS SOFTWARE 20% OFF CATALOG PRICE
CASHIERS CHECK OR MONEY ORDER MUST ACCOMPANY ALL ORDERS

(817) 825-4027

NOCONA ELECTRONICS • BOX 593 • NOCONA, TX 76255



This month we bring you the first installment of an occassionally-published series of portable computer programs designed by Fred B. Scerbo. While these programs will be designed with the professional educator in mind, they will be perfectly suited for use by parents who wish to help their children improve their grades and studies. If you have any ideas or needs you would like to see developed in this series, submit your requests to Fred, c/o PCM. We don't promise any results, but your suggestion might serve as the stimulus for a program for the Model 100 or 200.

For those of you who are readers of PCM's sister publication, THE RAINBOW, I will probably need little introduction. I am a public school special needs teacher in Massachusetts and have been writing programs using Tandy computers for the better part of the last five years. While the majority of my programming has been done on Tandy's highly successful Color Computer and Model III/IV, for the past year and a half I have been experimenting with a 24K Model 100. While my first love will always be the Color Computer with its graphic capabilities, I have found myself using my portable computer, or PoCo as it has come to be known, more and more every week.

This month, I will offer a short program which will help teachers and parents prepare written tests for school or home with little effort. This program, *Multiple Choice Test Maker*, or TESTM.BA as you could name it in memory, will free you up for the more personal contact that education requires while still allowing you to create professional looking paper tests.

Why the PoCo?

I first purchased my Model 100 over a year ago at the sale price of $799.95 at the local Radio Shack. When the machine first came out, I thought I would never possibly have any use for such a computer since my Color Computers usually met the needs of my students. However, all of that changed in March of 1984 when I was asked to help do the pairings for kids at a

---

(Fred Scerbo is a special needs instructor for the North Adams Public Schools. He holds a master's in education and owns his own software firm, Illustrated Memory Banks, in North Adams, Mass.)

state freestyle wrestling tournament. After spending hours doing the pairings by hand, I came to the conclusion that the task could probably be done much more efficiently by computer.

To make a long story short, I developed a program for my Color Computer to handle the task, which it did quite nicely. The one problem I faced was the CoCo's lack of portability. (I usually have a three hour drive to each tournament.) The logical answer was to buy a Model 100, which I very quickly fell in love with.

Since that time, I have gone to tournaments all over the country bringing along my Model 100 and dazzling everyone with what my little 24K battery operated wonder can do to save them time. However, that is not the only use I have found for my PoCo. I have written every one of my RAINBOW articles on it for the past year, and using a null modem, have dumped the contents to a Color Computer disk for delivery to the fine folks at Falsoft. There is just something very relaxing about sitting in my easy chair and constructing my column using the PoCo's built-in word processor. (In fact, I'm doing that this very minute.) Therefore, it was simple translating some of my educational programs for the CoCo to the PoCo so parents and my fellow educators could have the same convenience in preparing materials for school or home.

A Philosophy of Education

The most abused word in education today is CAI which stands for Computer Assisted Instruction. Somehow, a number of administrators and well-meaning parents and politicians have gotten the idea that the best way to improve education in our schools is to have every public school student seated in front of a computer monitor doing school work. As I outlined in previous RAINBOW articles, the most important area of CAI is probably the most ignored, that is, the use of computers to create paper test and quiz material for classroom use. This is the greatest *assistance* that a teacher can have since teaching is a hands-on experience which requires all the creative talents a teacher can muster. Since preparation of quiz and test material is a tedious time-consuming activity, using a computer to solve this task will free up the instructor for more creative efforts.

That is why I originally created this program for the CoCo. However, since our little PoCo is just as capable of dumping its contents to the line printer as the CoCo, I figured why not give it a try.

The Program

TESTM.BA differs from its Color Computer version in that it lets you use the Model 100/200's built-in text processor to create the information used in the written test. I had originally used DATA statements in the CoCo version since most individuals find it easier to edit a DATA line than to edit a file. Since our PoCo uses a full screen cursor to edit text, there was no need to waste memory using DATA lines in BASIC.

The kind of test that TESTM.BA makes is a multiple choice quiz allowing five answers including "Not Given" as a choice. Unlike other tests which would require you to create one correct answer to a question and three incorrect responses, TESTM.BA chooses its incorrect responses from the correct responses to other questions or terms. The order of every selection is different each RUN as is the order of the choices and selection of incorrect responses. Therefore, if you enter the material you want tested *once*, you can create an endless number of tests, all different, in as little time as it takes to load the file.

"Teaching is a hands-on experience which requires all the creative talents a teacher can muster."

Another nice feature of TESTM.BA is that it allows you to select print sizes: either your standard 80 character/line or the enlarged 40 character/line. If your line printer does not recognize CHR$(31) for enlarged or CHR$(30) for standard print, then change the value of the variable PS in lines 360 and 370 to the correct values for your printer. The program will adjust variables for the different sizes of print and allow you to control how much text is printed on a sheet as you approach the bottom of a page. You will be prompted to press 'N' for the next line of text only, or ENTER should you wish to print another whole page. This is especially useful if you are near the bottom of the page and are in the middle of a question which you do not want to split between two pages.

ROMEO.DO TEXT

"A plague o' both your houses! I am sped.", "MERCUTIO to everyone curses both families for being mortally wounded."

"This day's black fate on mo days doth depend; This but begins the woe others must end.", "ROMEO TO BENVOLIO — This looks like the start of something bad."

"O' I am fortune's fool!", "ROMEO TO BENVOLIO — Fate is having a good time making my life difficult."

"I beg for justice, which thou, Prince, must give.", "CAPULET'S WIFE — begs Prince for justice by killing Romeo."

"A gentler judgement vanished from his lips — Not body's death but body's banishment.", "FRIAR TO ROMEO — states that the Prince could have killed Romeo but chose instead to banish him."

"Thou canst not speak of that thou dost not feel.", "ROMEO TO FRIAR — states Friar cannot know how he feels about love because he is a man of the cloth."

"This is dear mercy, and thou seest it not.", "FRIAR TO ROMEO — states Romeo doesn't appreciate the fact that the Prince has banished him rather than put him to death."

"Stand up, stand up! Stand and you be a man, for Juliet's sake, for her sake, rise and stand.", "NURSE TO ROMEO — criticizes Romeo for his attitude and tells him to straighten up for Juliet needs him."

"Therefore have done. Some grief shows much of love, But much grief shows still some want of wit.", "MOTHER TO JULIET — states it is OK to cry for the dead, but too much makes you look foolish."

"Indeed, I never shall be satisfied, With Romeo till I find him dead.", "JULIET TO MOTHER — states she is angry enough to possibly kill Romeo for killing Tybalt."

"Soon asleep in quiet. O, how my heart abhors to hear him named and cannot come to him.", "JULIET TO MOTHER — states she would poison Romeo if she could get her hands on him, but he is not here."

"I'll to the Friar to know his remedy. If all else fail, myself have power to die.", "JULIET TO NURSE — states she will seek the Friar's help or kill herself."

"END", QUOTES FROM ROMEO and JULIET

Preparing a Text File

I have included a listing of twelve quotes from *Romeo and Juliet* and their interpretation which you may ENTER to see how the test prints out with a test which fills two pages. This is useful so you can see how the page advance works.

The text is set up in this fashion:

"Quote from play.","What it means."

Each question is thus composed of two parts: a question or term and its answer or definition. Thus, to avoid confusion, wrap each piece of information in quotation marks as you see above. Separate the question/answer or term/definition combination with a comma outside the quotation marks. This allows the computer to differentiate between the two pieces of information without being confused by commas used for punctuation inside the quotes.

I will give you a small sample demonstration that I have used in other programs to demonstrate this. Using the *TEXT* editor, open a file called SAMPL.DO for our text. We will create a short test on antonyms or opposites. Type in this information exactly as you see it below:

```
"UP","DOWN"
"HOT","COLD"
"LATE","EARLY"
"LIGHT","DARK"
"ALWAYS","NEVER"
"YESTERDAY","TOMORROW"
"END","SAMPLE TEST OF OPPOSITES"
```

To make TESTM.BA work, you must have at least five sets of terms/questions to use. The last piece of information will be "END","TITLE OF QUIZ." This assures that our title appears at the top of the quiz.

Return to the BASIC program after you have completed your text preparation. When prompted to load a text file, load SAMPL as the title. Do not add the .DO ending as the program will do that for you. After the file has loaded, you will be asked if you wish to reverse the data in the program. This would allow you to ask Down to be matched with Up, instead of Up matched with Down. This allows you to create two totally different kinds of tests with just one text entry. Next you may select large or small print. Prepare your line printer and press ENTER to start printing the test.

As you approach the bottom of the page, the screen will prompt you for additional lines or printing another full page. If you do not have enough text to fill a page, then the program will return to the titlecard. You may then reload the file and print another, different version of the same test material.

Other Uses

I have tried to keep TESTM.BA as small as possible so you could leave it in memory as a BASIC program. That way, you would only need to create new files of test data as needed.

The possibilities for TESTM.BA are only limited by your imagination. Create a fill-in sentence with the corresponding answer:

"Our first . . . was George Washington.", "president"

Use it with math problems:

"2 + 2 = ","4"

I'm sure your can think of other ways to use it as well. Just remember, try not to have the same answers for different questions, such as:

"2 + 2 = ","4"
"3 + 1 = ","4"

This could cause you to have an answer repeated which might appear confusing when the test is printed.

Errors

One word of warning: Take great care in the preparation of your text that you only use one comma for each pair of information. If you get an ?EOF error you probably left out a comma or added an extra one. You can avoid this problem altogether by pressing ENTER between the term and definition and leave out the comma altogether. You be the judge of which works better.

Also, you may have to CLEAR more string space if you get an ?OS error. Simply increase the value after the CLEAR statement at the beginning of the program.

Conclusion

I hope TESTM.BA and ROMEO.DO help you get a good idea how to use this program. In coming issues I'll offer other educational aids which I have developed. Let me know what you think.

PCM BAR CODED LISTING

The Listing:

```
1Ø REM MULTIPLE CHOICE TESTMAKER
2Ø REM BY FRED B.SCERBO (C)1985
3Ø REM 149 BARBOUR ST.N.ADAMS,MA Ø1247
4Ø CLEAR6ØØØ:DIMAO(5Ø),A$(5Ø),B$(5Ø),NP(
5Ø):GOTO11Ø
5Ø IFLEN(JK$)<=SWTHEN9Ø
6Ø FORT=SW TO ØSTEP-1:IFMID$(JK$,T,1)="
"THEN8Ø
7Ø NEXTT:GOTO9Ø
8Ø L$=LEFT$(JK$,T):W$=L$:GOSUB1ØØ:JK$="
    "+RIGHT$(JK$,(LEN(JK$))-T):GOTO5Ø
9Ø W$=JK$:RETURN
1ØØ LPRINTW$:CR=CR+1:GOSUB62Ø:RETURN
11Ø REM START
12Ø CLS:PRINT@47,"MULTIPLE CHOICE TEST M
AKER"
13Ø PRINTTAB(18)"BY":PRINTTAB(12)"FRED B
. SCERBO":PRINTTAB(1Ø)"COPYRIGHT (C) 198
5"
14Ø PRINT@248,"ENTER FILE NAME:";:INPUT
GG$:GG$=GG$+".DO"
15Ø CLS:PRINT@131,"NOW LOADING FILE";:OP
EN GG$ FOR INPUT AS1
16Ø FORJ=1TO5Ø:IF EOF(1)THEN2ØØ
17Ø INPUT#1,A$(J),B$(J)
18Ø IFA$(J)="END"THEN2ØØ
19Ø NEXTJ
2ØØ CLOSE1:PT$=B$(J):J=J-1
21Ø CLS:PRINT@124,"WANT TO REVERSE THE D
ATA (Y/N) ?"
22Ø T$=INKEY$:UH=RND(6666):IFT$="N"THEN2
6Ø
23Ø IF T$="Y"THEN25Ø
24Ø GOTO22Ø
25Ø CLS:FOR Q=1 TO J-1:TEM$=A$(Q):A$(Q)=
```

```
B$(Q):B$(Q)=TEM$:NEXTQ
26Ø J=J-1
27Ø FORI=1TO J
28Ø AO(I)=INT(RND(1)*J)+1
29Ø IF NP(AO(I))=1THEN28Ø
3ØØ NP(AO(I))=1:NEXTI
31Ø CLS:PRINT@124,"SELECT (L)ARGE OR (S)
MALL PRINT";
32Ø P$=INKEY$:IF P$=""THEN32Ø
33Ø IFP$="L"THEN 36Ø
34Ø IFP$="S"THEN 37Ø
35Ø GOTO32Ø
36Ø SW=38:PL=37:TL=8:LL=21:PS=31:GOTO38Ø
37Ø SW=76:PL=54:TL=14:LL=53:PS=3Ø
38Ø CLS:PRINT@124,"PRESS <ENTER> TO BEGI
N PRINTING";
39Ø P$=INKEY$:IFP$=CHR$(13)THEN41Ø
4ØØ GOTO39Ø
41Ø LPRINTCHR$(PS);" NAME";STRING$(LL,95
);"DATE";STRING$(TL,95)
42Ø LPRINT" TEST ON: ";PT$:LPRINT" ":CR=
4
43Ø FORP=1TO J
44Ø IFP<1ØTHEN ID$=" "ELSE ID$=""
45Ø LPRINT" ":CR=CR+1:GOSUB62Ø:JK$=ID$+S
TR$(P)+". "+A$(AO(P))+" "+STRING$(6,95)+
"  ":GOSUB5Ø
46Ø FOR Q=1TO4
47Ø C(Q)=INT(RND(1)*J)+1:IF C(Q)=AO(P) T
HEN47Ø
48Ø FOR K=Q-1 TO ØSTEP-1:IF C(K)=C(Q) TH
EN47Ø
49Ø NEXTK
5ØØ NEXTQ:C(5)=AO(P)
51Ø FOR E=1TO5
52Ø F(E)=INT(RND(1)*5)+1
53Ø FOR K=E-1 TO Ø STEP-1:IF F(K)=F(E) T
HEN52Ø
54Ø NEXTK:NEXTE
55Ø JK$=JK$+" (1) "+B$(C(F(1))):GOSUB5Ø
56Ø JK$=JK$+" (2) "+B$(C(F(2))):GOSUB5Ø
57Ø JK$=JK$+" (3) "+B$(C(F(3))):GOSUB5Ø
58Ø JK$=JK$+" (4) "+B$(C(F(4))):GOSUB5Ø
59Ø JK$=JK$+" (5) NOT GIVEN":GOSUB5Ø
6ØØ LPRINTJK$:CR=CR+1:GOSUB62Ø
61Ø LPRINT" ":CR=CR+1:GOSUB62Ø:NEXTP:RUN
62Ø IF CR<=PL THEN RETURN
63Ø CLS:PRINT@86,"ADVANCE PAPER TO NEXT
SHEET":PRINTTAB(7)"PRESS <ENTER> TO CONT
INUE"
64Ø PRINTTAB(6)"PRESS (N) FOR NEXT LINE
ONLY";
65Ø P$=INKEY$:IFP$=CHR$(13)THEN CR=Ø:RET
URN
66Ø IFP$="N"THENRETURN
67Ø GOTO65Ø
```

PCM

The Preble NavComp II
Two Aviation Programs in one package!
New Enhanced Version For the Model 100 or new Tandy 200!

NavPln
For Flight Planning

- Creates Navigation Log!
- Saves Route of Flight for repeat use!
- Compensates for Climbs and Descents!
- Statute or Nautical measurements!
- Self Prompting with error prevention!
- Fly direct or use VORS!
- And many other advanced features!

NavAid
For In Flight Navigation

- Tracks your flight in Real Time!
- Modify Flight Data as necessary!
- Incredible Graphic displays!
- Shows Position, Fuel, Intersections!
- ETA'S, Ground Speed, and much more!
- Turn off and on without data loss!

Requires Radio Shack Model 100 with at least 16K or the new Tandy 200!

Attention Radio Amateurs! Announcing:
The Preble HAMTERM!
For the Model 100 or new Tandy 200!

- Radioteletype operation for BAUDOT, ASCII, AMTOR, CW!
- Split Screen Operation in receive mode with unlimited type ahead capability!
- Word Buffer in Transmit Mode! Corrects errors before they are sent!
- Automatic CQ! — Automatic Station Call! — Automatic Logging!
- Echo to printer! — Download! — Upload! — Send Time and Date! — Send RYRY!
- And many other advanced features!

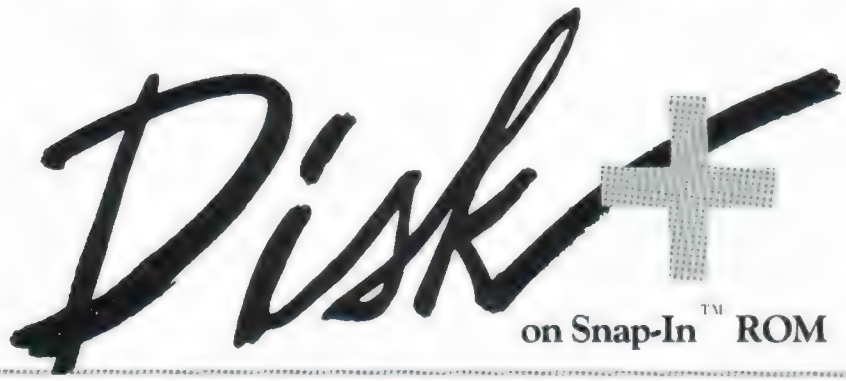
Requires Kantronics UTU, RS Model 100 with at least 16K or the new Tandy 200!
The Preble NavComp: Only $34.95
The Preble HAMTERM: Only $29.95
Shipping and handling $1.50 to U.S. and Canada — $5.00 to other countries
Free Brochures available!
VISA, MasterCard or COD is OK!
Order From: Dr. Preble's Programs; 6540 Outer Loop; Louisville, KY 40228 (502) 966-8281

TRS-80™
SAVE SALES TAX *PLUS DISCOUNT
*TEXAS RESIDENTS ADD ONLY 4⅛%

TANDY 200-72K

FORT WORTH COMPUTERS
WE ARE SERIOUS ABOUT SAVING YOU MONEY
(Located 30 miles from Fort Worth)
377 PLAZA • GRANBURY • NR FORT WORTH, TEXAS 76048
CALL FREE: 1-800-433-S-A-V-E
Monday thru Friday - 9:00 a.m. to 5:00 p.m.
(Order inquiries/Customer Service &) IN TEXAS: 817-573-4111

USE YOUR DESKTOP COMPUTER AS A DISK DRIVE FOR YOUR M-100.



Try *Disk+* for 30 days. If you aren't as excited as we are, return it for a full refund.

When we designed *Disk+* we did it out of necessity. We wanted a way that we could just connect a Model 100 to our desktop computer with a cable and save files onto the desktop's disk drive. We wanted it to be so simple to use it would be self-explanatory.

Picture this. *Disk+* comes to you on a Snap-in ROM and a diskette for your desktop. You take a quarter and open the little compartment on the back of your Model 100. Then you just press the ROM into the socket. *Disk+* appears on your main menu just like a built-in.

You connect your Model 100 to your other computer using an RS232 cable (available from PCSG for $40).

You just place the *Disk+* diskette into the desktop's drive and turn on the computer. It powers up automatically and says "awaiting command" on your desktop's screen. Then you just put the widebar cursor on the Model 100 main menu on *Disk+* and press ENTER. You are shown your RAM files arranged just like the main menu.

To save a file to your other system's disk drive, you just move the widebar cursor to the file you want to save and press ENTER. It is saved instantly with no further action.

To look at the disk directory, you just press a function key on your Model 100. You see immediately the disk directory on your Model 100 screen, and it is arranged just like your Model 100's main menu.

To load a file from the diskette to your Model 100, you just move the widebar cursor to the file and press ENTER. The file is transferred to your Model 100's RAM instantly. You can press F8 and go back to the main menu, and the file you loaded from diskette is there, ready to use.

It is so nice to be able to keep your documents, programs (both BASIC and machine code) and *Lucid* spreadsheet files on the diskette, and bring them back when you need them. All files are ready to run or use with no changes or protocol by you.

If you have access to a desktop computer and don't have *Disk+*, then evidently we have done a poor job telling you about it.

All files and programs that you load or save, go over and come back exactly as they are supposed to be because of full error checking. This guaranteed integrity is really a comfort. *Disk+* is wonderful in so many other ways. For example, you can do a "save all" of all your RAM files with just a touch of a function key. That group of files is saved on the diskette under a single filename with a .SD (for subdirectory) extension. Any time you want, you can bring back all those files at once, or just one or two if you like, again with one-button ease.

Disk+ takes up no RAM. That's zero bytes either for storing the program or for operating overhead.

What really excites most *Disk+* users is text file cross compatibility. Your Model 100's text files are usable on your desktop computer, and your desktop's text files become Model 100 text files.

This means you can write something on your Model 100, and with *Disk+* transfer it instantly to your desktop and start using it right away on your bigger computer. Or the way we like to work is to type in a document on the desktop computer and then transfer it to our Model 100 with *Disk+*. Then we print out the document, beautifully formatted, using WRITE ROM.

Disk+ works with just about every micro sold, from IBM PC and its clones, to all Radio Shack computers (yes, all), to Apple II, Kaypro, Epson and most CPM. Just ask us. More than likely, your computer is supported.

Incidentally, hundreds of Model 100 owners have gone to their Radio Shack stores and bought a color computer because it is so low priced, and with *Disk+* they have an inexpensive disk drive.

And if that weren't enough, how about this: *Disk+* also provides cross-compatibility between different computers like IBM, Apple or the Model 4 using the Model 100 as the intermediary device. Quite a feature!

The snap-in ROM is really great because you can use other ROMs like *Lucid* or WRITE ROM. They snap in and out as easily as an Atari game cartridge and you never lose your files in RAM.

Anyone who ever uses *Disk+* simply can't do without it. But so many times we have had new users call us and say, "Wow! I had no idea when I ordered it that *Disk+* would be so fantastic. I just couldn't believe that I could use my desktop computer's disk drive with my Model 100 just like it is another main menu."

That's why we sell *Disk+* on a thirty-day trial. If you aren't completely satisfied, return it within thirty days for a full refund. Priced at $149.95 on Snap-in ROM. MasterCard, Visa or COD.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd., #207, Dallas, Texas 75229 • 1(214) 351-0564

PCSG provides hotline software support for the Model 100. Call us at 1-214-351-0564

© PCSG 1985

As our initial dBASE II programming project, we have chosen this highly useful mailing list system. In doing so, we acknowledge that old postal admonition about what it is that 'must go through'. . .

dMAIL: Project One

By Danny Humphress
PCM Technical Editor

This month, as promised, we begin working on a dBASE programming project — a mailing list program. The program will have fields for name, street address, city, state, ZIP code and telephone number. It will allow the user to add and edit records, print a mailing list, and print mailing labels.

Once you've decided what you want the program to do, the next step is to define the database files that will be used with it. In the case of our mailing list program, there will only be one file. The file will have an index on name for looking up records and printing alphabetical lists, and an index on the ZIP code to be used for printing mailing labels in ZIP code order to take advantage of pre-sorted postage rates.

Our database file will be called MAIL.DBF. The alphabetical index will be MAILNAME.NDX, and the ZIP code index will be MAILZIP.NDX. It is a good idea to use some common word or first letters, such as "MAIL," when naming databases and indexes. As the directory becomes full with files, this makes it easy to see which indexes go with which database files.

The structure for MAIL is seen in Figure 1. Use dBASE's CREATE command to set up the file.

Now, create the two index files from the dBASE dot (.) prompt with the following commands:

```
USE MAIL
INDEX ON NAME TO MAILNAME
INDEX ON ZIP TO MAILZIP
```

To keep things as simple as possible, the program will not use a lot of advanced dBASE techniques — just straight, simple programming. Many of the commands used are ones that we've already discussed in this series. Others are new and we'll go over them as they arise.

We will have four different dBASE programs which will make up the mailing list system: MAILMENU.PRG, MAILEDIT.PRG, MAILREPT.PRG and MAILLABL.PRG. The first program, MAILMENU.PRG, will be the main menu for the system and the other three will be called from it. MAILEDIT.PRG will be the program used for adding and editing records. MAILREPT.PRG and MAILLABL.PRG will be used for printing the mailing list report and the mailing labels. A screen format file, MAILEDIT.FMT, will be used by MAILEDIT.PRG.

Figure 1

```
STRUCTURE FOR FILE:  C:MAIL     .DBF
NUMBER OF RECORDS:   00000
DATE OF LAST UPDATE: 08/01/85
PRIMARY USE DATABASE
FLD       NAME       TYPE WIDTH    DEC
001       NAME       C    030
002       STREET     C    030
003       CITY       C    015
004       STATE      C    002
005       ZIP        C    010
006       PHONE      C    013
** TOTAL **                00101
```

The first program that we'll work on will be MAILEDIT.PRG — the one essential program and the most complicated. Since it will use the MAILEDIT.FMT screen format file, we will also have to create it.

MAILEDIT.FMT

Figure 2 contains the screen format file, MAILEDIT.FMT. Enter it with a word processor, MS-DOS's EDLIN, or dBASE's MODIFY COMMAND line editor.

As you can see, the screen format file is a simple list of @ SAY/GET commands with PICTURE clauses used where needed. This screen is already designed for you, so you don't have to worry about making it look nice. When designing your own screens, it's a good idea to lay it out either on paper or with a word processor, then figure out the screen coordinates and create the necessary @ SAY/GET command lines.

The PICTURE clause for the STATE field (!!) converts whatever two characters the user enters into uppercase to conform with the two-letter state abbreviations.

The ZIP field is pictured to accept either the old five-digit ZIP codes or the new ZIP+4 nine-digit codes. A space is used to separate the two numbers instead of the usual hyphen (-). If we had not done this, five-digit ZIP codes would have a hyphen after them. This way, there is only a harmless space.

The TELEPHONE field's PICTURE clause is set up to accept the area code and automatically put in the punctuation marks (parentheses and hyphen).

MAILEDIT.PRG

Type in the program in Figure 3, MAILEDIT.PRG, exactly as it appears. Again, you may use a compatible word processor, EDLIN or MODIFY COMMAND to do this. The comments in italics are for your use only and should not be typed in. Additionally, the line numbers are for reference and should not be entered.

You may have noticed the use of upper- and lowercase words. A common practice when programming in dBASE is to put all the dBASE commands and keywords in lowercase and the fields, variables and filenames in uppercase. While not mandatory, this makes it easier to read and understand the program when the time comes to debug or modify it.

The first four command lines in the program set up the operating conditions for dBASE.

SET TALK OFF tells dBASE to refrain from displaying operating messages and record numbers. It is often the first line of any dBASE program.

dBASE normally puts colons before and after each field when doing full-screen editing. SET COLON OFF turns off this feature.

SET CONFIRM ON tells dBASE to wait for the user to press ENTER before going on to the next field when doing full-screen editing. If confirm is on, dBASE will automatically advance to the next field if you fill up a field.

SET BELL OFF tells dBASE to be silent when you enter the last character in a field. Like the previous two commands, this is just a matter of preference.

Line 6 is the familiar old USE command. We are opening the MAIL.DBF file with the indexes MAILNAME.NDX and MAILZIP.NDX. Since MAILNAME is the first index listed, it will be the one used for the FIND command. MAILZIP is kept updated, though, and will be used for the mailing label program.

The screen is cleared in Line 8 and we are ready to begin.

Figure 2

```
@ 10,15 say 'Name:    ' get NAME
@ 12,15 say 'Street:  ' get STREET
@ 14,15 say 'City:    ' get CITY
@ 14,39 say  State: ' get STATE picture '!!'
@ 14,49 say 'ZIP: ' get ZIP picture '99999 9999'
@ 16,15 say 'Telephone Number: ' get PHONE picture '(999)999-9999'
```

Remember the DO WHILE/ENDDO command pair? We discussed it in the May dBASE Tutor. Since T means "always true" in dBASE and F means "always false," the command in Line 10, DO WHILE T, will run forever. Forever, that is, until we RETURN back to the dBASE dot prompt.

Lines 12 through 18 use a simple @ SAY/GET to find out which function the user wants to use. Since we are getting data into a variable and not a field, we have to use the READ command in Line 16 to invoke the @ SAY/GET command. The next line, CLEAR GETS, makes dBASE forget about the previous @ SAY/GET command lines so it does not try to read them again with READ, APPEND or EDIT. Line 18 clears the line where the message appeared.

The commands in lines 20 and 24 are probably new to you, but they are quite easy to understand. IF ANSWER='A' tells dBASE to do the following commands up to the ENDIF if the content of the variable ANSWER is equal to 'A.' If the expression ANSWER='A', were false, dBASE would skip down to the next line after the ENDIF. This section is the one used when the user presses 'A' to add records.

Line 21 opens our screen format file,

Protect Your Valuable PCM Collection With . . .

DISTINCTIVE AND DURABLE PCM BINDERS

These distinctive brown binders not only ensure that your magazines stay in mint condition, but they showcase your collection as well. Each binder is clearly embossed with the magazine's name in gold lettering on both the front and the spine. They're a handsome addition to any decor.

Each of these handsome binders holds 12 issues of PCM and is only $7.50 (please add $2.50 for shipping and handling for each binder ordered).

SPECIAL DISCOUNTS ON PAST ISSUES WITH THIS OFFER

If you place an order for six or more back issues of PCM at the same time you order your binder(s), you are entitled to $1 off each back issue of the magazine, which normally sell for the single issue cover price. For an index to our back issues, and an order form, please refer to our "Back Issue Information" page (check Table of Contents under departmental listings).

YES. Please send me _____ sets of binders at $7.50 each (plus $2.50 shipping and handling for each binder ordered). If your order is to be sent via U.S. Mail to a post office box or to another country, please add $2. Kentucky residents add 5 percent sales tax.
I also want to take advantage of the special savings of $1 off the single issue cover price for back issues with the purchase of a binder. (An order form from a recent issue indicating the back issues you wish to receive should accompany this order.)

Name _____

Address _____

City _____ *State* _____ *ZIP* _____

_____ *My check in the amount of* _____ *is enclosed. (In order to hold down costs, we do not bill.)*
Charge to: ☐ *VISA* ☐ *MasterCard* ☐ *American Express*

Account Number _____ *Expiration Date* _____

Signature _____

Mail to: Falsoft, Inc., The Falsoft Building, 9529 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059. **PCM**

MAILEDIT.FMT, for the next command.

Line 22 performs the APPEND command using the current full-screen editing format file.

When dBASE comes across this command, it will display an empty record on the screen using the format we set up. It will allow the user to enter as many records as he likes until he presses CONTROL-Q on an empty record.

The next command, Line 23, closes the format file. This keeps it from popping up when we get back to the READ command in Line 16.

Line 24, of course, terminates this IF section of the program.

We begin another IF section in Line 16. This time, we're seeing if the person pressed 'E' to edit records. If he did, dBASE moves on to Line 27 and does the processing related to editing records, otherwise it skips down to 45.

Lines 27 through 30 ask for the name the user is searching for and stores it in the SEARCH variable.

The extra spaces after SEARCH are trimmed off in Line 31. This keeps the user from having to enter the whole name. If you were searching for the name, PCM Magazine," you could just enter PCM. Otherwise, if you just entered PCM with no TRIM function, dBASE would look for the word PCM followed by 27 spaces. That is, there would have to be a record with nothing other than PCM in the NAME field.

The command in Line 32 uses the index to find the name entered into the SEARCH variable. The ampersand (&) preceding SEARCH must be used because we are looking for the word contained in SEARCH and not for the word "SEARCH" itself. This is called a macro — a subject for a later article.

We begin another IF conditional in Line 33. We check to see if the record number is zero. If it is, then the FIND command failed to locate the entered name and an error message needs to be displayed. If it is not zero, the process continues with Line 40 — after the matching ENDIF in Line 39.

Lines 34 through 38 are executed if the record is not found. A message is displayed and the computer waits for the user to strike a key. The command SET CONSOLE OFF in Line 35 prevents dBASE from displaying a "WAITING" message while executing Line 36. The display is turned back on in Line 37. Because the record was not found, we want to go back to the top of the program and wait for another command. The LOOP command in Line 38 takes us back up to the nearest DO WHILE command.

In Line 40, we set the screen format to MAILEDIT.

We edit the current record (the one we just found) with EDIT # in Line 41.

Line 42 closes the screen format file.

If the person had selected 'X' to exit the program, it would be detected in Line 45 and the following two statements would be executed.

Line 46, a solitary USE command, closes the database file, and Line 47 returns to the dBASE dot prompt. If this program had been called from another, as it will when we are using the menu program, control would return to that program.

The last line, Line 50, is the ENDDO command that takes us back up to the DO WHILE command in Line 10 and then on to wait for another command.

This program can be run by entering DO MAILEDIT at the dBASE dot prompt, or by entering DBASE MAILEDIT at the MS-DOS prompt.

Next

There have been a lot of new commands and techniques thrown at you in this month's "Tutor," but seeing how the commands work in a real program makes them much easier to comprehend.

Experiment a bit with the program if you like, but keep a copy of the original around because we'll be using it next month as we add the report and mailing label programs. Have fun!

Figure 3

```
 1  set talk off
 2  set colon off
 3  set confirm on
 4  set bell off
 5
 6  use MAIL index MAILNAME,MAILZIP
 7
 8  erase
 9
10  do while T
11
12     store ' ' to ANSWER
13     erase
14     @ 20,25 say 'Select: [A]dd [E]dit e[X]it ';
15     get ANSWER picture '!'
16     read
17     clear gets
18     @ 20,0
19
20     if ANSWER='A'
21        set format to MAILEDIT
22        append
23        set format to
24     endif
25
26     if ANSWER='E'
27        store '          (30 spaces)              ' to SEARCH
28        @ 20,19 say 'Enter name: ' get SEARCH
29        read
30        clear gets
31        store trim(SEARCH) to SEARCH
32        find &SEARCH
33        if #=0
34           @ 21,18 say 'Name not found.  Press any key to continue.'
35           set console off
36           wait
37           set console on
38           loop
39        endif
40        set format to MAILEDIT
41        edit #
42        set format to
43     endif
44
45     if ANSWER='X'
46        use
47        return
48     endif
49
50  enddo
```

PCM

We're Talking Graphics Transmission — Get the Picture?

By Randy Graham
PCM Contributing Editor

Graphics is one of the fastest growing aspects of personal computing. Graphics show up everywhere. Businesses use them to make the charts and graphs which brighten up reports. Computer-assisted designing and other simulations integrate vast quantities of data into easily grasped concepts. And where would TV commercials be without graphics?

So, you are into graphics. You have a really nice application and with the aid of a graphics program, you create a real masterpiece. Naturally you save it to disk or tape. But a graphics design is like a party dress; how often can you wear it again? Wouldn't it be nice if you could share it with friends or even sell it to stangers for re-use! And, as a matter of fact, before you start creating that masterpiece to fit the particular need, why not ask whether just the right piece is already available in some library? You know what we always say about reinventing the wheel.

Well, you can always put copy disks in a library somewhere and list them in some catalog. And, on request, you can mail a disk to a friend in another city. But you know good and well that in this department the questions are: Can graphics be sent over transmission networks like other data files? Are there accessible libraries? Where and how? The answers to the above are: yes . . . yes . . . however . . .

Think TV. A beam in the picture tube is scanning the optical image focused on its face. At each spot, it modulates a voltage proportional to the brightness of that spot, from zero to total. In its most elementary form, if the spot's brightness is over its design "threshold," it sends a positive voltage; below the threshold there is no voltage. At the end of a line, it shuts off its sensing to re-trace to the start of the next line (no bidirectional scanning here). At the end of the last line, it shuts off to return to the top of the screen to trace lines between the first set.

The receiver replicates this process, sending a pulse or no pulse to agitate the phosphor on the face of the picture tube and thereby reproduce the visual image. For this system to work, camera and receiver must be synchronized to trace and re-trace together. The information stream is therefore composed of voltage-no voltage plus synchronizing signals.

Think computer. The information stream is a series of voltage-no voltage, on/off, yes-no, high-low, X's and O's, ones and zeros. Hey, this is a bit stream, and what we have here is a long string of binary numbers zipping around the cables and airwaves! And computers know what to do with binary numbers: read them, manipulate them, store them. In fact, if you read the bits in groups of eight, you can convert them to Hex digits and store them byte by byte. This process of reading a bit stream and converting it into Hex digits for processing and storage is called "digitizing" — the most popular word in electronics today.

Keep thinking computer. It has a circuit to read the bit stream from one of the CPU's output ports and convert it into a TV signal which is sent to your monitor screen. This circuit gets its information from a reserved section of memory which holds enough digital data to fill the screen. In computer graphics we talk about reading the screen or dumping the screen to a printer. Not so. We are reading and writing to the screen memory section.

This column is not about graphics; it is about databases. I just want to make sure you understand digitizing a TV signal so that it can be processed, stored and retrieved. Our description was of the most rudimentary systems used mostly in security cameras and so forth. A real TV signal contains a variable value for each pixel's brightness to achieve something of a gray scale and a color code for the pixel, so that each pixel requires a whole byte or even a 16-bit word. And of course, in "real TV" an audio signal will be added, etc.

OK, we can draw pictures with our graphics program and save it in digital form. Now, about those online libraries. Right now you will find three kinds of graphics in libraries of information services and bulletin boards.

The simplest form seems downright primitive just five or ten years into the Computer Age. Back in the days when the only form of data transmission was the teletype, operators used to while away the idle hours sending pictures to their buddies down the line. Since all they had were capital letters on a roll of paper, the pictures were made with just X's and blank spaces. The more creative used other keyboard characters for shading. Secretaries used to do this on their typewriters, too, and I still call it "typewriter art." This mechanical sort of art form still exists in databases and is an easy way to get started. The pictures are not intended for screen display — you dump them right to your printer. I suppose that if you are using an 80-column screen you can display them and see a miniature version.

A more sophisticated picture file is one that reads the TV signal in digital

---

(Randy Graham is a rehabilitation counselor working with the handicapped. Personal computing is his hobby; telecommunications, one of his favorite activities. He has done freelance information retrieval and is an inveterate user of the major online systems.)

form, stripping off all synchronizing codes, etc. The program is fairly simple and you will find them in public domain in graphics libraries. The important thing is that the program must run on your system, since you are leaving ASCII behind and talking directly to your CPU, the topic of conversation being its memory map. You must also have a similar little program that can read the data files and convert them back to a picture which shows on your screen. Again, you can find simple ones in public domain.

These picture files will usually be BASIC DATA statements and may be in two forms. They may be self-contained with the graphics generating program appended, so when you load and run it, the picture appears on your screen. Or, the file can just be a series of numbers so that you load the program, run it, tell it to read the file and make the picture. Either way, the data file is usually in ASCII format.

Now, these files have a couple of characteristics. One is that they are very long. The old Chinese proverb that a picture is worth a thousand words proved they invented computers centuries ago. Remember that when you get ready to download a picture file. You must have quite a large memory buffer to receive it. I mean, we are talking multi-K's here. And the length of time it takes to download them may make you think again about going for a 1200-Baud modem.

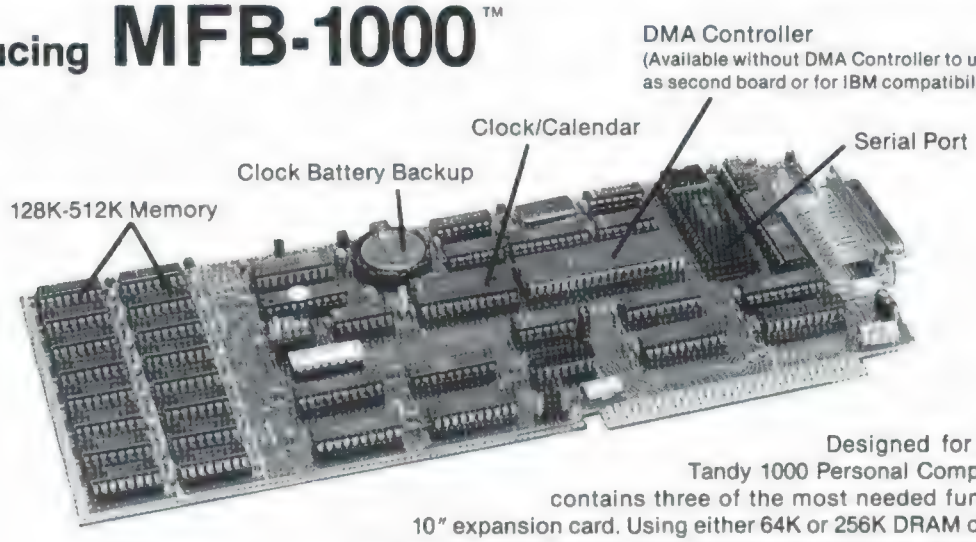
The other characteristic is that since the bytes are read serially, if there is any break or distortion in transmission, you get nothing but garbage when you try to make the picture. The best defense is to use a "bulletproofing" protocol like XModem in your terminal program. ASCII files are pretty safe, but this becomes even more important when we go to binary files.

The third kind of file is used by a really fancy store-bought graphics program which can produce beautiful Hi-Res pictures with shading and colors, etc. The data file is so big that it is usually stored as a binary file. Now, you can send binary files overline, but it can be a little tricky. Your terminal program must support it and an error-checking protocol is essential. The results are worth it.

If you want to explore this fascinating — even addictive — area of personal computing, CompuServe is a good place to start. We portable people can find graphics files in our forum, although I must confess that I am not much taken by the tiny little pictures that will fit on the LCD display. Of course, the 100/200 is perfectly capable of downloading files for programs which will generate print graphics. You MS-DOS folks can look in the IBM forum and find a whole section on computer art. CompuServe has its own "bombproof" terminal program and it checks to see if you are running it; if not, it will not download its binary files to you. If you think you may spend much time in their graphics libraries, you might just want to purchase one for your machine from them. It is a good all-round terminal program as well as having a high reliability for binary files.

Ready to jump into graphics? You will need a graphics program which will run on your system, a terminal program and modem which supports an error-checking routine like XModem and plenty of memory. Now, start playing around. Experiment. Start sharing with others in the field. Just be ready. Someday the boss is going to say, "Who can get me a nice graphic to brighten up this presentation?" Raise your hand. Even if you are the boss. Or, do I have to draw you a picture? PCM

Introducing MFB-1000™



Designed for use with the new Tandy 1000 Personal Computer, the MFB-1000 contains three of the most needed functions on a single 10" expansion card. Using either 64K or 256K DRAM chips, the MFB-1000 can be populated with up to 512 K of memory bringing total system memory to 640K. Additionally, the board also includes an IBM compatible serial communications port (identified as COM1) and a battery backed real time clock/calendar. As required by the design of the Tandy 1000, the MFB-1000 also contains its own DMA Controller. However, the DMA Controller can be disabled, making the MFB-1000 compatible with the Tandy 1200 as well as other IBM compatible machines.

pbj inc.
P.O. Box 813
911 Columbia Avenue
N. Bergen, N.J. 07047
(201) 330-1898
Dealer Inquiries Welcome

■ 128K RAM $329.95 ■ 256K RAM $369.95 ■ 512K RAM $429.95

ORDERING INFORMATION:
Visa/Mastercard orders accepted. Allow 2 weeks for personal checks to clear. Add $3.00 shipping and handling. C.O.D. orders add $2.00. N.J. residents add 6% sales tax.

IBM is the registered trademark of International Business Machines
Tandy 1000/1200 are registered trademarks of Tandy Corp

MEGABYTES FOR MINI-BUCKS

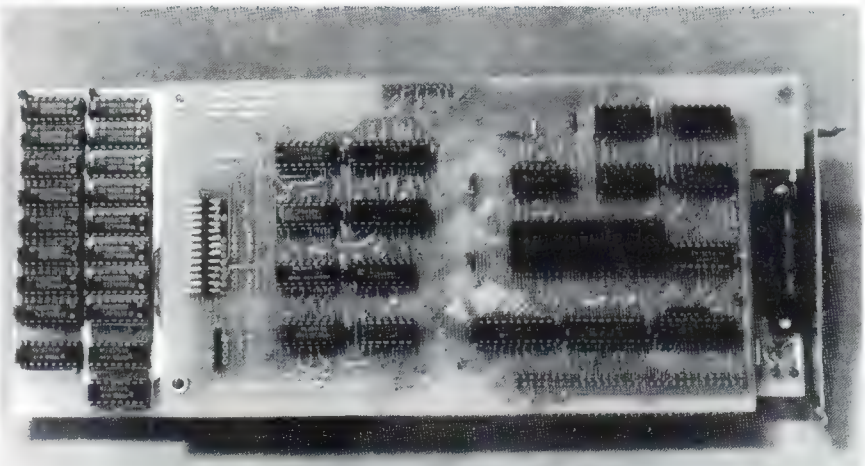
Tandy 1000 Hardware

Tandy 1000

TanPak

The first and only board that your Tandy 1000 may ever need. Your 1000 is very versatile, new applications and functions are being developed for it every day. And now with the HDS TanPak you can keep your options open for tomorrows technology. As you know the Tandy 1000 only has 3 expansion slots, and those are not quite PC compatible. The TanPak does the job using one expansion slot that normally would require four using Tandy boards. The TanPak includes a Serial Port, a Clock, and sockets for up to 512K of Memory Expansion. And the best part of all, it will save you some big bucks over the Tandy boards.

TanPak 128K ..... **$399.**



Tandy 1000

TanTel

300/1200 Baud Internal Modem

Both 300 and 1200 Baud within the 10" that is alloted by Tandy for a board. But that's not all. This internal modem also features automatic dialing, an auto-answer feature, full or half duplex, automatic speed selection, and a speaker to alert you to busy signals and wrong numbers. FCC registared and full documentation included.

Internal Tandy 1000 Modem "TanTel" ... **$299.**

AS SEEN AT THE RADIO SHACK COMPUTER SHOWCASES

Disk Drives

Fast stepping add on or replacement disk drives for half the Radio Shack price ..... **$149.**

Tandy 1000

Eprom Programmers

These EPROM Programmers plug into the Model 1000 edge card slot and complete with software and documentation. The 24 pin programmer will program the following EPROMS: 2704, 2708, 2716, 2732, 2508, 2516, 2532, 68764, and 68766. The 28 pin programmer will program the following EPROMS: 2764, 27128. and 27256.

24 Pin Programmer ..... **$149.**
28 Pin Programmer ..... **$149.**

Hard Drive Kit

Put a 10 Megabyte Hard Drive inside your Tandy 1000. This Hard drive subsystem includes an interface card and a Half-Height Hard Drive that directly replaces a floppy disk drive in both size and power consumption drive.

10 Megabyte Internal Drive System ..... **$895.**
10 Megabyte External Drive System ..... **$1095.**
15 Megabyte External Drive System ..... **$1395.**
30 Megabyte External Drive System ..... **$1895.**

Radio Shack and Tandy are trademarks of Tandy Corporation, TanPak, and TanTel are trademarks of Compukit Corporation

HARD DRIVE SPECIALIST

Ordering Information
Use our Watts line to place your order Via Visa, MasterCard, or Wire Transfer. Or Mail you payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. Most items are shipped off the shelf with the exception of hard drive products that are custom built. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request

1-713-480-6000
Order Line 1-800-231-6671
16208 Hickory Knoll,
Houston, Texas 77059

XTREE Utility Grants Extra Power To MS-DOS

By Lonnie Falk

MS-DOS is such a powerful operating system that it is easy to understand why it has won such great acceptance in the world of microcomputers. So widely is it used that it has, for all practical purposes, become pretty much the *de facto* standard in the personal computer field.

A couple of the concepts, however, are not quite so easily understood. Chief among them, especially to newcomers, is the directory tree concept — the possibility of having subdirectories within other directories and that various files can be allocated in each of these directories.

Then, there are things you just cannot do. One of them, particularly galling, is the inability to rename a directory. I remember when I set up a file for depreciation and capital assets and named it TEECEE (after the initials of our accountant) while I was developing it. When it was finished, however, it seemed more appropriate to call it ASSETS — but there was no way to rename a directory.

Finally, there is the inability to do very much with more than one file at a time. This, primarily, is why TEECEE still exists as a directory. I know I could have created a new directory named ASSETS, moved all the files in the TEECEE directory, one by one, to ASSETS, and then deleted all the files from TEECEE and, finally, removed TEECEE. It seemed easier to just get the people in bookkeeping used to using TEECEE as the directory where the assets programs were stored. I know our CPA feels justly honored!

Despite the brilliance of Bill Gates and his minions at Microsoft, there are a lot of things that *could* have been done with MS-DOS. Fortunately for all of us, the people at Executive Systems, Inc. seem to have done them.

They have created a fine utility package called *XTREE* which should be at the top of just about anyone's hit parade. *XTREE* solves all the problems I mentioned above and a lot more, besides. In addition, it does it with elegance and style that is certain to make it a "must" for just about any MS-DOS user.

Let's go into some detail here:

The main *XTREE* screen gives you five important types of information — and gives them to you the way you want it. At the very top of the screen (which, by the way, uses colors very effectively) your current path is shown. This is significant since the "main" part of the screen is a graphics representation of the tree structure of your disk — showing the main directories, subdirectories and any directories under that. These are arranged in alphabetical order (a nice touch) and you can use the arrow or PG UP and PG DN keys to quickly move through all the directories you have.

As you move through the directories, a highlight bar shows where you are and the path indication changes at the top of the screen. You always know not only what directory you are in, but how to get there. I have used this part of the display as a teaching aid here in the office to explain the directory tree structure to a number of our people.

Below the graphics representation of the tree structure is a listing of the files in the selected directory, once again in alphabetical order. You access the window by simply pressing the RETURN key and then use the PAGE or arrow keys to scroll through these files.

To jump ahead, a more complete listing of files is available simply by pressing RETURN again. The graphics representation of the directory tree goes away and even more files can be shown.

To the right of the short file window and the graphics tree are a set of windows which give statistics on the disk you have accessed and the files. This is very helpful information.

I suppose I should mention at this point that *XTREE* is fast! It runs through an entire 10 megabyte hard disk on a Tandy 1000 or 1200 in about five seconds. For the Tandy 2000 it takes about double that time since it uses an alternate screen drawing module which is somewhat slower. Even with the 2000 version, I have no complaints as to speed. The 1000/1200 version is pretty blinding when you consider *XTREE* is looking at 10 million bytes.

Were this all *XTREE* did, I would be moderately pleased that it allows me to find things so readily on my disk. However, we've really only touched the tip of this marvelous utility.

XTREE's true power comes from its ability to manipulate files and directories. You are able to delete, make and rename directories at the push of a single button. Additionally, you can show all files in a directory (or specify certain files) in an expanded window, log onto another disk or determine the available file space for making copies.

Let's look at that last one for a minute. Suppose you need to know whether you have space on a floppy (or a hard disk, for that matter) to copy a set of files. Instead of adding bytes with a calculator and then seeing how much vacant space there is on a disk, you simply use the Available command to check the space. Because *XTREE* also allows you to specify a group of files and find the total number of bytes in the group, it is a simple matter to determine whether you have space.

Oh yes, there is also a Volume command so you can change the name of the disk without reformatting it. Super, super, super.

I have saved the best part of this directory command explanation for last — the tagging process. By specifying a certain kind of file, you can tag them for other functions. This, like all of *XTREE*'s commands, is a simple one-button operation.

But, there is more: file commands which allow you to set attributes, copy files, delete files, rename files and view files. In combination with the tag functions, which work on files as well, you can copy whole groups of files with a simple keystroke.

It is here that the real power of

XTREE comes into play. Want to copy all your .COM files from the root directory to a new BIN directory? Simply use a single keystroke to tag all the .COM files, a second to copy them to the new directory and a third to delete them from the root directory.

The view files function lets you look at the file itself, in either ASCII or hexadecimal format. You can mark special parts of the file if you wish, and even control the speed the file scrolls on the screen. You can also move through the file a line at a time by hand if you wish.

Finally, *XTREE* has the ability to exit to DOS to perform other functions or run other programs. Once you have done that, you simply return to *XTREE*.

XTREE has fine error-handling and is extremely easy to work with. It is not copy-protected, so you can move it to your hard disk without any problem. And, as mentioned earlier, it is extremely fast and very easy to use. We had it up and running about five minutes after we opened the box. Menu bars at the bottom of the screen virtually eliminate the need to use the manual and there is online help as well.

Utility packages should make life easier for the person using them. *XTREE* easily does this and does it extremely well. If you are going to purchase a utility to help you view, manipulate or restructure files or directories, *XTREE* is absolutely for you. It is a beauty of a program!

(Executive Systems, Inc., 15300 Ventura Blvd., Suite 305, Sherman Oaks, CA 91403, $49.95)

SOFTWARE

TuneSmith: 'Composer and Record Album' for the 1000, 1200

For those who thought they had everything for their computer, how about *TuneSmith*, a "record album"? Now, we can enjoy songs played on our computer like "Clair de Lune," "Malaguena," "What is This Thing Called Love," "Foggy Mountain Breakdown" or classical selections from Bach.

TuneSmith runs on the Tandy 1000 and 1200, and their clones. The songs play on the 2000, but the tune editor does not work. The sound generator on the 2000 makes the Bach music seem like it's played on a harpsichord. The software operates in color on an RGB monitor, but unfortunately is copy-protected. It's unusual for a fifty dollar software package to be copy-protected. The program disk must be resident in Drive A on the hard disk Tandy 1200. The ten songs included with *TuneSmith* are written in BASIC, so they can be copied to another computer, such as the 2000, and run from BASIC.

This copy-protection causes problems on the Tandy 1000, since not all floppy disk drives are made by the same source. The *TuneSmith* music editor would not work on several machines at the Radio Shack computer store because the copy-protection scheme won't read a non-standard drive.

The software editors are trying to revise the program to make certain it works on all 1000s. A good benchmark test for your 1000 is to try *TuneSmith*.

NEW CUSTOM SOFTWARE for your MODEL 100 or NEC PC 8201A

CBUG

CBUG is the ultimate debugging utility for your lap computer. Requires less than 3K RAM. 100% machine language. Features include:

- Instruction Step
- Call/Instruction Step
- Zero/Fill Memory
- Load Data
- Alter Registers
- Dump Memory & Registers
- Search/Replace
- HEX math (+ & −)
- Decimal — HEX conversions
- Display in HEX or ASCII

$29.95

RENUMBER

The utility you've been waiting for! Now you can renumber your BASIC programs IN PLACE! This powerful, 100% machine language program is relocatable to any convenient place in memory. Requires less than 3K RAM. Renumber all or part of a program. You specify line numbers and increment.

$24.95

LET'S PLAY MONOPOLY*

It's you against the computer and the computer is a tough competitor. Super fast machine language graphics display the whole board at all times. The computer makes all its own decisions. You're really going to love this game. $29.95

ASSEMBLER

A powerful 8085 assembler for your lap computer. Recognizes all the standard pseud-ops. Immediate data may be in decimal, hex or ASCII. Relocatable to any convenient place in memory. Several built-in Macros make your programming easier. FAST! Written in 100% machine language and requires less than 3K RAM.

$32.95

MELODY MAKER

A musical program generator. Use simple cursor controls to enter sheet music Melody Maker writes a program to play the song. You can even use Melody Maker to add musical routines to your own programs. $19.95

CUSTOM SOFTWARE

• 1308 Western • Wellington, Kansas 67152 •

316/326-6197

VISA

Please include $1.00 for shipping. For orders outside the U.S., include sufficient postage for airmail delivery. U.S. funds only.

MasterCard

*Monopoly is a trademark of Parker Brothers

If the program works, the drives are standard, so other software should work with no problem.

The software is packaged exactly like a record album, with a 22-page user's manual in the album jacket. Besides a selection of songs to play on the computer, there is an on-screen tutorial, a demo program showing differences in tempo and other enhancements of "Red River Valley," a help screen and a full screen editor.

When songs are played, a piano keyboard is illustrated at the bottom of the screen, and both treble and bass clefs are shown at the right-hand side, with the note(s) moving along the score.

Forty function keys are used to edit the songs and increase or decrease tempo, raise or lower octave, play or edit music, change playmode and select color monitor background, foreground and border. Word processing functions control the cursor, erase and insert charaeters and move between bars and measures in the song.

It is a bit more difficult to compose music using BASIC play statements, rather than seeing a musical score on screen and choosing notes like the freeware program *PC Musician* does. However, if it's been a long time since you had piano lessons, and if you are composing music by ear rather than from sheet music, the play statement data feature shouldn't bother you.

A nice feature of this software is the ability to program two "voices" to play at the same time. The Tandy computer soundboard has one voice function, but *TuneSmith* figured out a way to enter chords that play along with the melody line. The lower notes are programmed to sound a fraction of a second before the higher notes, and that fools the listener's ear into hearing both the melody and the accompaniment at the same time.

A useful section in the software manual is instructions on how to wire an external speaker to the computer. There is also a schematic diagram to connect the computer to a stereo amplifier, using $2 worth of parts from Radio Shack. Other attachments to the computer, such as delays, phasers or distortion boxes can be added to make the computer sound like a rock band.

There's a good music reference section in the user's manual that reviews sheet music symbols. BASIC play statements such as note or pause lengths, tempo, and music foreground and background are explained very clearly. Programming chords or adapting scores is simple with *TuneSmith*, and the manual has excellent instructions.

This music program will amaze your friends, and is a nice break from business software. Perhaps Mozart could have written his music a lot faster using *TuneSmith*.

— M.J. Batham

(Blackhawk Data Corporation, 307 N. Michigan Avenue, Chicago, IL 60601, (312) 236-8473, $49.95)

SOFTWARE

DATA/SORT Sorted Databases with User-Definable Forms

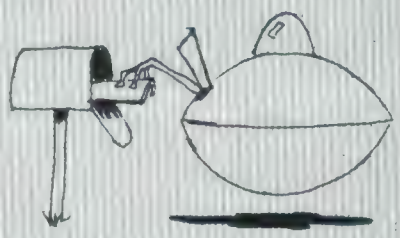
By Jay Halcomb

DATA/SORT is an attractive, easy-to-use and efficient tool. The package creates and maintains sorted database files in RAM, and can send database records to a variety of destinations: RAM text files, other computers (directly or via phone lines), the printer, or the cassette recorder (or copy the entire database to disk via the disk-video interface). One can merge databases with form letters to print the ubiquitous personalized form letter. Finally, users can design their own data entry forms, and enter data manually or with the bar code reader.

The product comes in the usual Tandy packaging: a vinyl case containing a cassette with the necessary files (nine of them), an instruction manual of 42 pages and a pocket reference guide. The manual is well-done, and covers the steps of loading and running the programs very explicitly. As a result, the program seemed quite easy to use upon first running it, despite a few fumble-fingered mistakes on my part.

Of the nine files on the cassette tape, the user will normally be concerned only with two programs: either *DATA* (about 6.5K) or *DATAW* (about 7.5K), and *SORT* (about 2.5K). The other files are example files, and programs con-

PCM



Back Issue Availability

Back copies of many issues of PCM are still available.

All back issues sell for the single issue cover price. In addition, there is a $2 charge for the first issue plus 50 cents for each additional issue shipped in the U.S. When possible, issues are shipped UPS. The postage cost in Canada and Mexico is $3 for the first issue and $1 for each additional issue.

VISA, MasterCard and American Express accepted. Kentucky residents please add 5 percent state sales tax. We do not bill and no COD orders accepted.

We suggest you order back issues you want now while supplies last.

To order, just fill out the form on the next page and mail it with your payment to:

PCM
The Falsoft Building
Prospect, KY 40059

PCM

☐ Please send me the following back issues:

(Payment must accompany back issue orders.)

NO.	MO.	YR.		PRICE	
			VOLUME 1		
1	JULY	'83	PREMIER ISSUE	$3.00	☐
2	AUG.	'83		$3.00	☐
3	SEPT.	'83		$3.00	☐
4	OCT.	'83		$3.00	☐
5	NOV.	'83		$3.00	☐
6	DEC.	'83		$3.00	☐
7	JAN.	'84		$3.00	☐
8	FEB.	'84		$3.00	☐
9	MAR.	'84		$3.00	☐
10	APR.	'84	LIMITED	$3.00	☐
12	JUNE	'84		$3.00	☐
			VOLUME 2		
2	AUG.	'84		$3.00	☐
3	SEPT.	'84		$3.00	☐
4	OCT.	'84		$3.00	☐
5	NOV.	'84		$3.00	☐
6	DEC.	'84		$3.00	☐
7	JAN.	'85		$3.00	☐
8	FEB.	'85		$3.00	☐
9	MAR.	'85		$3.00	☐
10	APR.	'85		$3.00	☐
11	MAY	'85		$3.00	☐
12	JUNE	'85		$3.00	☐
			VOLUME 3		
1	JULY	'85		$3.00	☐
2	AUG.	'85		$3.00	☐

TOTAL _____
SHIPPING & HANDLING _____
KY RESIDENTS ADD 5% _____
TOTAL AMOUNT ENCLOSED _____

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
☐ ☐ ☐ ☐ ☐ PAYMENT ENCLOSED
CARD # _____ EXPIRATION DATE _____ PHONE # _____
SIGNATURE _____

cerned with loading a small assembly language program called by the main programs.

The principal database program (written in BASIC) comes in two versions: *DATA* and *DATAW*. *DATAW* is a duplicate of *DATA*, except that it allows the use of the bar code reader. The other main program is *SORT* (also in BASIC), which, of course, sorts the database files the user creates in RAM using *DATA* or *DATAW*. Since *DATAW* is a clone of *DATA*, I will describe *DATA*'s functions in some detail, and note later the added capabilities which *DATAW* supplies through the use of the bar code reader.

DATA creates and maintains alphanumeric database files in RAM, in ASCII format. Upon first running *DATA*, the user is asked for the name of the form file he or she wants to use. The form file (which must begin with 'D') is a small RAM file which users create, using *TEXT*, and which simultaneously defines the data record (fields) and data entry form (fields and prompts) for a particular application. (Three sample forms, called 'D' screens, are provided: DNOTE, DINV and DADRS.)

A data record can use up to 16 fields of variable length, and the size of the record is limited only by the constraint that one complete record, including prompts, must occupy only one Model 100 screen (eight lines). The prompts appear only when the user accesses the database through *DATA*; they are not written to the database as part of the record.

After supplying the name of the desired data form, the user is asked for the name of the data file he or she wishes to access. When the user has supplied this, the display is filled with the data form previously named, with blank fields. At this point the user can do several things (the function keys, with their labels displayed on the label line, provide the branches):

1) Begin entering data into a form. The data (alphanumeric) can be edited with the DELete key, and the ENTER key moves one from field to field. When the user has entered all the data he or she desires, there is the option of appending the record to the named database and beginning a new record, or exiting to menu and discarding the record.

2) Review all the records, or a selected subset, in the named database. One can search in any field with any specified search string, and all the records matching the search string will be consecutively displayed at the user's prompting. An old record can be edited and resaved.

3) Output the database to a device or other RAM file. The output format can be specified in a variety of ways: columnar or spread, all fields or selected fields, one or several lines, all records or selected records. The device can be the cassette recorder, a COM line (which might be attached to another computer, or a phone line) or the printer. All of the database, or selected records, can be appended to any existing text file to create reports.

4) Merge the database with a previously prepared document stored as a .DO file in RAM, such as a form letter. With this capability one can merge all records or a selected subset, and choose the fields to appear in the form letter. The output from the merge can be sent to another RAM file, the cassette, the printer or a COM line.

The *SORT* program offers the ability to sort the selected database on any given field, with three types of sort: distinguishing uppercase from lowercase, not distinguishing uppercase from lower, and numeric sort (which sorts numbers absolutely, regardless of the digit spaces they occupy in the field being sorted on; e.g., 001 and 1 are treated alike). To run *SORT* one first supplies the names of the data form and the database; then the user is given a display of the data form, and chooses the field to sort on by moving the cursor with the cursor keys.

DATAW operates just like *DATA*, except that within the data form files — the 'D' screens — the user can indicate the fields to hold bar code input. Then, with *DATAW*, the user can enter information into those record fields with the wand, or enter information manually (and enter data into the non-wand fields manually as usual). This requires that the user have his or her own bar code reader software, which *DATAW* uses. *DATAW* accepts the widely used decoding programs: UPC2, 3 of 9, Plessey (and perhaps others, though which ones aren't indicated). I did not run *DATAW*, as I lack a bar code reader.

DATA and *SORT* performed all their functions quite rapidly, with no difficulties. The creation of the data forms took a (very) little trouble as one has to be a bit careful in the format.

But once the initial learning phase was passed, the creation of forms to taste was no problem. The initial user should have no problem running *DATA/SORT* to its full capabilities within a couple of hours of reviewing the manual. The program offers a strong selection of capabilities, and is well-designed. The real point of the Model 100, of course, is that it is truly a portable computer. Anyone who has the need to collect formatted data in the field will find *DATA/SORT* well suited to his purpose. With the ability to send the database records to a larger computer, formatted as the user likes, the 100 can be used for the field collection, and later manipulate the data as desired with the user's primary computer.

The manual intimates that the small assembly language subroutine, called by *DATA* and *SORT*, will not be a menu selection, but this is not the case. The routine itself is loaded just below high memory, where it takes just a couple hundred bytes, but if you have another program which wants that space, there are simple arrangements for lowering the routine.

The *DATA* program seems to require about 2.5K of free space available in RAM to run. If it doesn't find it, the message "Memory Full" is displayed, and the program shuts down so that the user can free some space by deleting files or copying them to another device. The *SORT* program doesn't require any extra free space, as it sorts the database within its file.

Redirecting any of the output from *DATA* from one device to another requires that the user reformat the data each time, even if he or she wants to use the same format as previously used. This is a minor inconvenience, as formatting the data is quickly done by answering a short series of questions.

One can't directly edit a database using *TEXT* (unless with extreme care), as this is liable to change the size of a record field, and make the database forever after inaccessible to *DATA*. But, of course, old data records can always be accessed and edited through *DATA*, for updates.

(Tandy Corp., One Tandy Center, Ft. Worth, TX 76102, $49.95)

SOFTWARE

Synonym Finder: Words For The Wise

Synonym Finder by Writing Consultants, is a single disk program designed to give a writer an online dictionary of over 90,000 words with 9,000 keywords that average over 10 synonyms each. This program integrates itself nicely with *WordStar* or *Multimate*. The program takes up very little disk space and since it's only called when required, very little memory is used.

When the package arrived I was a little disappointed, at first, with the documentation. The first thing I thought when I saw the photocopied booklet was, "Another thrown together software package." On the first page I read, ". . . all you need to operate it is this letter." Well, they are right. You load it first and then your word processor. The only time you know it's there is when you need it. This package is so easy to use that anyone, in a matter of minutes, can check the spelling and

SOFTWARE PRICE WAR

INTRODUCTORY PRICE

$49.95

(REGULAR $99.95)

30 Day Money Back Guarantee

Powerful & sophisticated payroll program Yet Easy to use on your IBM Computer (or compatible)

Complete Employee Records. All government required information Detailed 941 form, W2's, FICA, FWT, SWT (for your state) & misc. deductions

The automatic mode in this system can be set up in less than one minute to produce up to 999 pay checks

Imagine - from the first key stroke to the last in less than one minute

SAVE TIME - SAVE MONEY - ORDER NOW
This revolutionary breakthrough in payroll systems

Joe Lynns Payroll System 49.95
Shipping & Handling 3.00
5% Service Charge for Credit Cards (2.50) ..... ____
TOTAL AMOUNT OF ORDER ..... $ ____

MasterCard Joe Lynn VISA
Computer Services
23501 W. Gagne Ln. • Plainfield, IL 60544
815-436-4477

MSDOS UTILITIES
TANDY 1000, 1200, 2000
IBM PC/XT/AT

No Technical Knowledge Required !!! Easy to Use.

***** RAMDISK ***** $49

RAMDISK -- Use spare memory as a superfast disk to speed up applications. (You specify Ramdisk size). Appears to programs as a disk, but gives instant access. Easy to install and use.

***** ALPHA TOOL BOX ***** $59

ALPHABETIZED DIRECTORY DISPLAY -- View Directory on 1 screen (5 columns, sorted, Fast!).
FIND FILE -- Search all directories for file.
DIRECTORY DTREE -- Display all sub-directory names in an easily readable form.
BACKMOD MODIFIED FILES -- Quick, automatic backup of files. Backup in same format as original. Perfect for Hard Disks. Also for floppies.

***** BETA TOOL BOX ***** $59

SINGLE KEY COMMAND -- Define any text string (or DOS Command) as a single keystroke. Press single key to execute commands!
COUNT LINES -- Superfast count of lines in a file.
DUMP FILE IN HEX -- Formatted dump of file in Hex/Ascii. Scroll Forward, Backward, Begin, End.
CHANGE FILE ATTRIBUTES -- Make files hidden, readonly, archive, etc. Set or reset attributes.

***** SERIAL PRINTERS ***** $49

Connect your Serial Printer to your RS232 port. Full handshaking. Easy commands. Complete instructions.

Don't miss the review of these ALPS utilities in the February, 1985 issue of 80-Micro (P.119).
More ALPS Tools Available. Call or Write for ALPS catalog. Customer Support !!!

ALPS
23 Angus Road
Warren, New Jersey 07060
201-647-7230

rearrange words in a document. This program is the "missing link" to a complete word processor.

To change words with *Multimate* is a two second operation. The ease of use is amazing. Take the statement "I was amazed to find this package so easy to use." to check the spelling and find a synonym for amazed, all you have to do is move the cursor to the 'a' in amazed and press CTRL/F6. You will be given a choice of words that look like this: "amaze: (vb) astonish, astound, dumbfound, flabbergast, surprise." Just move the cursor to the one you want and press ENTER. It will be inserted into your text. The only bug I've found is if you don't want to change the word and press ESC to cancel, the word disappears from the display. It's not gone from the document, just from the display. It will reappear when you make a page change or scroll the screen. This is just a minor irritation, but one that should be corrected.

I used this program with *WordStar* with the same outstanding results. This program is a must for the serious writer and a convenient and versatile addition to have for your word processor. The $124.95* price tag is a little steep for the hobbyist, but justifiable if you write frequently and, like me, need help with the English language.

*(*Editor's Note: Information at press time states that an introductory price of $79.95 is available to PCM readers, and that Writing Consultants has changed the program's name to* WordFinder.*)*

(Writing Consultants, 11 Creek Bend Drive, Fairport, NY 14450, 716-377-0130, $124.95)

— Brian Fernalld

SOFTWARE

Sargon III is a Royal Chess Mate

By Bruce Rothermel

Once upon a time there was a great Mesopotamian king who ruled Assyria. Although he was a kind and benevolent ruler, he was primarily known as a skillful warrior. He personally led his troops and often succeeded in winning battles where the numerical odds were against him. His name was Sargon II.

Today there is a *Sargon III*, not a king or benevolent ruler, but a very skillful opponent. *Sargon III* is the king of PC-DOS computer chess games — but runs royally on the Tandy 1000.

Marketed by Hayden Software, *Sargon III* requires only a 128K, single disk drive, monochrome monitor Model 1000 or 1200 to operate.

For those interested in learning chess, *Sargon III* may be a painless way to introduce yourself to the game without experiencing the frustration of having an experienced player coach you and constantly comment on your dumb moves. Although challenging at the upper levels, *Sargon III* also has many features to help a beginner.

The first chapter of the 85-page user guide has the information needed to get a novice to both the computer and chess running and playing. It includes instructions from the U.S. Chess Federation on the basic rules of chess.

Sargon III also has many options to make the program challenging to any player, from beginner to master. There are nine levels of play, ranging, in my opinion, from a challenging game to an impossibly ferocious opponent. An easy option stops *Sargon III* from thinking while you are thinking, making the odds more favorable for you.

Sargon III has these options: the ability to terminate the game currently in play and start a new one; to just quit when you can't take it any more; to change *Sargon*'s difficulty level while playing a game; and the ability to halt *Sargon*'s thinking at any time and force it to make a move. At higher levels, *Sargon III* is given more time to think and analyze the possible consequences of all possible moves and your counter moves. The more time given, the more difficult an opponent it is.

Also, other options include: Changing sides during a game. Once *Sargon III* has you hopelessly pinned, you can change sides and watch *Sargon III* wiggle out of the trap; inverting the playing board so your pieces are either at the top or bottom of the screen; taking back moves. This is an especially nice option that you rarely get with a human opponent. An especially bad move can be taken back even if the game has progressed quite a few moves. You can "unplay" the game one step at a time. You may have *Sargon III* suggest a move for you. Unlike a human competitor, this will be the best move *Sargon III* thinks you have. Truly a competitor without an ego.

You may have *Sargon III* act as a referee for two human players. *Sargon III* will check the legality of each move, maintain a move list and can replay, save or print the game that has been played.

Sargon III gives you the choice of turning the beeper on or off; replaying a game in memory; printing a list of moves in a game; writing the current board positions to a printer or saving a game to disk for continuation, analysis or replay later.

You may set up any position of pieces on the game board. You may change the position of pieces on the current game or set up a game in progress for analysis. You may also skip a move. It's cheating but you can do it.

Sargon III is well equipped to play a challenging game. It has an opening library of over 68,000 different positions. Occasionally you see *Sargon III* referencing its disk file before coming in for the kill. This opening library can be canceled, forcing *Sargon III* to think for itself. There is even an option to have *Sargon III* show you what it is thinking as it analyzes its possible moves. You also have the capability of using a light pen to move your pieces.

On the flip-side of the disk 107 historical chess games are stored. These were selected by American chess master Boris Baczynski. These games may be reviewed one move at a time or *Sargon III* will automatically replay the game.

In summary, *Sargon III* is an outstanding computer chess game. Sargon II would have been proud.

My only negative comment is that for beginning chess players, the easiest level is very difficult. A beginner will be annihilated by *Sargon III* unless he asks for a lot of help from *Sargon III* during the game.

In a Computer Chess tournament held in 1984, hosted by chess master Julio Kaplin, *Sargon III* easily beat out the other two top computer chess programs, *Blue Bush Chess* and *Spoc*. Kaplin, however was able to defeat *Sargon III*. Hurray for the humans!

If you want to sharpen your chess skills, take a break from number crunching, or just play with an opponent who won't smirk when your queen gets taken by a pawn, *Sargon III* might be

for you. But remember, Hayden Software is in the PC-DOS market and will not formally support Tandy computers.

(Hayden Software Company, 600 Suffolk Street, Lowell, MA 01854, $49.95)

SOFTWARE

Two Tender Morsels For Those On Byte Diet

By Dennis Kirley

The chief problem with your lap-top machine is memory: there is never enough of it. Text files devour it; BASIC programs even need work space. Each byte is a delicacy to be savored. (After all, they even call it a "menu.") To help you count bytes and prevent your programs from getting fat, Custom Software serves up two tasty machine-language morsels: *Men-U-Tility* and *ByteFyter*.

Both programs load slowly from cassette, POKEing large files into RAM. If you do not have other machine language files in your machine, loading is simple. But if you have other .CO files and need to relocate *ByteFyter* or *Men-U-Tility*, then you may find the documentation confusing. *Men-U-Tility* asks for the start address while *ByteFyter* insists on the highest address usable for BASIC. Since both programs come from Custom Software, the approach to relocating files ought to be consistent.

Men-U-Tility takes over the Model 100's menu to give you instant file length and function key utilities. It moves the display of available bytes to the top line of the LCD so that it can label the function keys. At the touch of a key, you can rename, kill, print or make files invisible. You can also set the day, date, time or even an alarm that works in all modes. *Men-U-Tility* lurks in protected memory, but 256 times each second your PoCo gives *Men-U-Tility* the opportunity to take control or sound the alarm. Best of all *Men-U-Tility* only uses 1,945 bytes and is very fast!

With *Men-U-Tility*, you can rename a file by highlighting it on the main menu and pressing F1. *Men-U-Tility* then prompts for the new filename. You can even rename the ROM programs. Killing a file is just as simple; select it and press F2. *Men-U-Tility* asks 'Sure?' and will only delete the file if you respond 'Y'. Want to make a file invisible to protect it from view or to save a menu slot? Select the file and press F4. If you need to make files reappear later, press CODE V.

Men-U-Tility will set day, date and time from the main menu sparing you the BASIC syntax. This is a nice but trivial feature, since BASIC is not that bad. In any case, unless you encounter frequent cold starts, you won't need to set the clock very often.

Much more useful is the alarm function which beeps no matter what you are doing. Press F8 from the main menu to set the alarm, and use BASIC, *TEXT* or *TELCOM*. You will hear the alarm no matter what program you are using. One complaint: when the alarm goes off, you must return to the main menu to shut it off. This is a real nuisance if you are online or in the midst of a program.

Men-U-Tility also formats and prints

text files. You can set a left margin, line width, top and bottom margin and page length. The print facility even lets you send embedded control codes to the printer for underlining, condensed print and the like. The only major print functions missing are headers and footers. *Men-U-Tility* is an excellent formatter for simple documents and drafts.

Although I did not test *Men-U-Tility* with the disk/video interface, the documentation indicates that the program will cause the menu to appear on whatever screen you are using, rather than forcing the menu to the LCD. This removes one of the major nuisances in using the interface.

Having a single machine language program is rarely a problem; adding a second can cause problems or even a cold start. Custom Software tries to eliminate this problem by making their files relocatable. Still, I did suffer cold starts using multiple .CO files. I strongly suggest backing up all files before loading a second machine language program.

ByteFyter is a good companion program to *Men-U-Tility*, but with a more limited audience. This machine language program removes unnecessary spaces and remarks from BASIC programs. Many packer programs will do as much. *ByteFyter* fights the byte by packing BASIC programs in place and even combining lines to save the five byte per line overhead. Like *Men-U-Tility*, *ByteFyter* is fast.

ByteFyter can be executed from the main menu, run from a BASIC program, or used as a subroutine. Once called, *ByteFyter* needs the name of the program and the maximum line length to use for combining lines. *ByteFyter* checks program logic before combining lines so that the compressed program will run correctly.

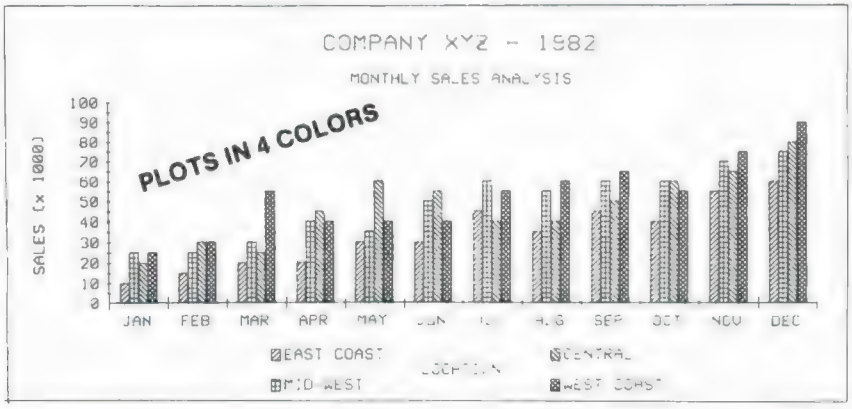
The manual cautions that you should copy each file to tape first since compression is done in place and the original file deleted. I did not encounter any problems, but this is good advice because certain BASIC combinations cause *ByteFyter* to choke. The compressed program may not run properly. Once finished, *ByteFyter* reports the original and the compressed size in bytes.

Custom Software indicates that *ByteFyter* is primarily for developers of BASIC programs. Running *ByteFyter* on purchased software saved some memory, but not much. The net result may be room to squeeze just one more program or text file into memory. Every byte counts, so even if you never write any BASIC, you may still benefit from a little byte fighting.

Men-U-Tility is billed as adding features to your portable that should have been there in the first place. Since I have kept a program for reporting file length and making files invisible on my machine almost as long as I have had it, I certainly agree. The added features are a real plus; *Men-U-Tility* allows you to manage every byte. It makes fighting the byte almost painless. The only thing it doesn't do is give you more memory. *ByteFyter* gives you the next best thing: more free memory. While your programs diet, you will be savoring each byte.

(Custom Software, 1308 Western, Wellington, Kansas, 67152, 316/326-6197. *Men-U-Tility* and *ByteFyter* supplied on cassette tape with documentation for $24.95 each plus $1 S/H)

Plot Against Your Competition!
with QuickPlot®



We're the people who gave the CGP-115 Color Plotter business status. With QuickPlot, your Model 100, or Tandy 200, produces high quality line-, bar- and pie-charts in vivid color. Simple data entry and flexible chart formatting makes QuickPlot and the CGP-115 Color Plotter the logical choice for the professional who wants to *plot* his profits—not *spend* them.

PCM CERTIFICATION SEAL

Complete QuickPlot Package Includes: Bar Chart, Pie Chart and Line Graph. Data and chart formats save to disk or tape for future plotting.

STILL ONLY
$49.00
We pay shipping on pre-paid orders

Model 100, Tandy 200 and CGP-115 are registered trademarks of Tandy Corporation.

Now **you** have a friend in the software business.

COMPUTER SOLUTIONS COMPANY

Computer Solutions Co.
P.O. Box 24383
Louisville, KY 40224-0383
(502) 245-0232

Call or write to place your order or send for free information.

Full-Screen Editor

Adam Systems Corporation has a new full-screen editor on the market.

PQfEDIT is a user-friendly full-screen editor program enabling you to edit a (ASCII) text file in a page rather than line-by-line format. It is an attractive alternative to the EDLIN line editor.

The program is ideal for anyone who needs to create, display, or update text diskette files for letters, charts, program data, documents, tables or programming language source statements such as PASCAL, FORTRAN, ASSEMBLER, C, and BASIC.

PQfEDIT is designed for the Tandy 2000, the IBM Personal Computers series and IBM compatibles running MS-DOS and with a 25-line by 80-column display. The program is compatible with any release of DOS (1.0 to 3.x).

Text is edited in a window of 23 lines by 75 or 80 characters. Editing features available include cursor positioning, character insert delete and replace, horizontal and vertical page scrolling, and commands such as delete, add, copy and move lines, join and split lines, locate and replace strings, and get and put lines to disk. The program can also accept embedded printer-formatting control codes.

PQfEDIT was written in assembler language. All 60 user commands are keyboard driven by a one- or two-key keystroke sequence. Each command is explained in the users' manual and summarized in both the on-line help display and the command reference card.

The program is provided in a portfolio to protect the system diskette and manual and comes with an unconditional 30-day guarantee.

PQfEDIT is $45 plus $2 shipping. Contact Adam Systems Corporation, 5919 Munson Court, Falls Church, Virginia 22041, (703) 3779-0669.

Local Area Network Enhancements

Standard Microsystems Corporation has added three application enhancement software packages to its local area network product line.

Standard Microsystems Corporation is marketing both the local and remote versions of *LAN: Mail Monitor*™, an electronic mail and file transfer system for up to 25 users on a local area network. The remote version includes software for communications over phone lines and provides for up to 10,000 users. In addition, the company will be offering *LAN: Datastore*™ with extended report, which is a local area network relational database for up to five simultaneous users. The package allows a central database to be shared among IBM PC's with complete data protection.

Standard Microsystems also designs and manufactures MOS/LSI and MOS/VLSI circuits as well as board level systems products used in computer peripherals, data and telecommunications equipment, home entertainment products and word processing terminals.

For further information, contact Lucy Tarnell, Standard Microsystems Corporation, 35 Marcus Boulevard, Hauppauge, NY 11788, (516) 273-3100.

Electronic Bilingual Pocket Dictionary

World travelers will never be at a loss for words again thanks to the TRANSLATOR 8000 electronic pocket dictionary. Marketed by Langenscheidt Publishers, Inc., the world's largest publisher of bilingual dictionaries, this state-of-the-art electronic device, compact enough for a handbag or pocket, is available in three editions: Spanish, French and German. Its suggested retail price is $69.95.

At the heart of this electronic dictionary is a microcomputer with 48K bytes of memory. Each version of the TRANSLATOR 8000, true to name, is programmed with 8000 words and phrases. Its bilingual memory produces 4000 words in English, and 4000 in the designated foreign language. Users can "call up" a word in either language at any time.

The business traveler will find the TRANSLATOR's 16-word memory bank helpful for storing and practicing jargon that applies uniquely to his business or industry. And every traveler can feel confident about mathematical calculations because the TRANSLATOR 8000 is also a calculator, and can be used to convert currency, metric measurements, the temperature or the dinner tip.

The bright yellow and blue TRANSLATOR 8000 is available nationally in computer, electronics, department and book stores as well as through some catalogs. Inquiries should be directed to Langenscheidt Publishers, 46-35 54th Road, Maspeth, NY 11378, (718) 784-0055.

Using *BAREAD 2.1*

Bar code listings must be read in numerical order beginning with Line 1 and continuing through the last line of the listing. The computer display is used to prompt you as to which line to scan and give you warning messages should you happen to get out of step.

When you run *BAREAD*, it asks you to scan the first line of the bar code listing. This line contains the name of the program as well as the beginning of the program itself. The computer will sound a high-pitched beep whenever it's ready for you to scan a line. After a line has been successfully read, you'll hear a lower beep. A "blip-bloop" sound prompts you to turn your attention to the screen for a message. You'll hear this when you accidentally scan a line out of sequence.

After reading the first line, you continue scanning with the second line. Remember to wait for a high beep before scanning and then listen for a low beep to indicate a successful read.

Once the last line of the listing has been scanned, *BAREAD* will return control to the Tandy 100/200 menu screen. Note that the program you just scanned is now in the directory with a .DO extension.

The final step is to convert the .DO text file to a normal BASIC program. This is done quite simply by going to BASIC and loading the file with a command such as LOAD"TEST .DO" (if the program name were TEST). The program will load into BASIC and will be ready to run. To save the program in BASIC's compressed format (.BA extension), you'd type SAVE"TEST" (if the program were named TEST). You may then kill the .DO file with KILL "TEST.DO".

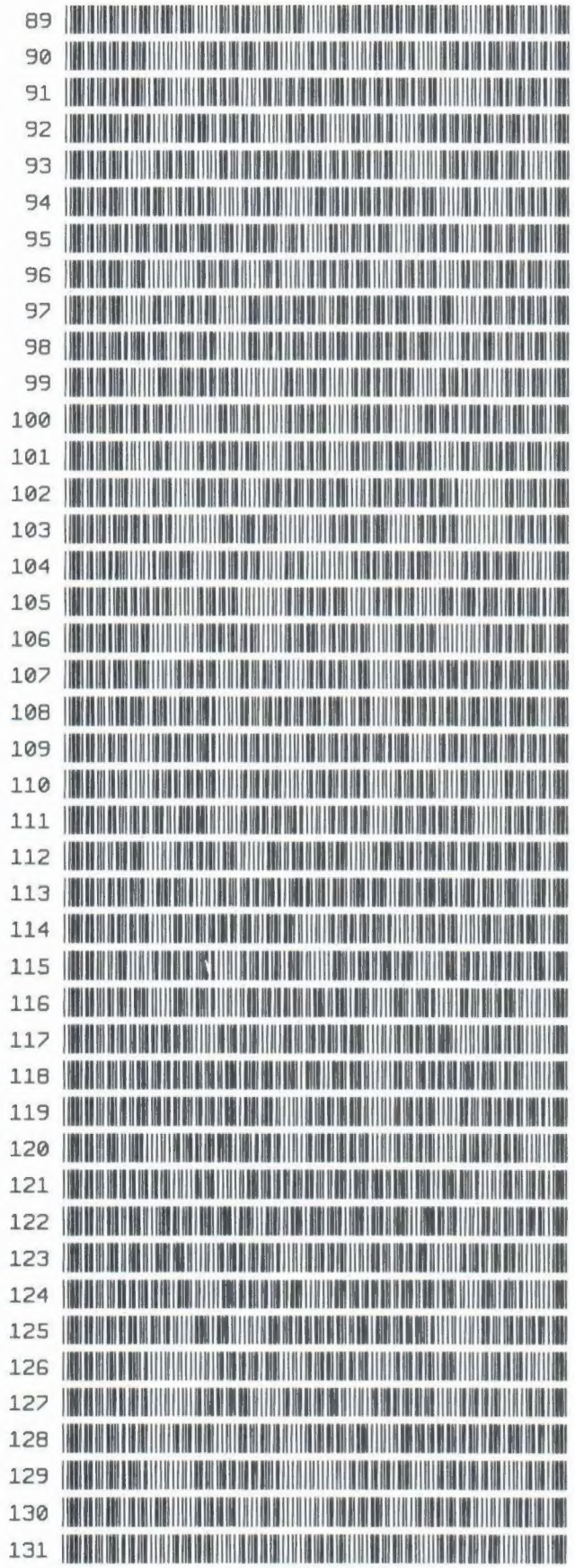
BAREAD Version 2.1

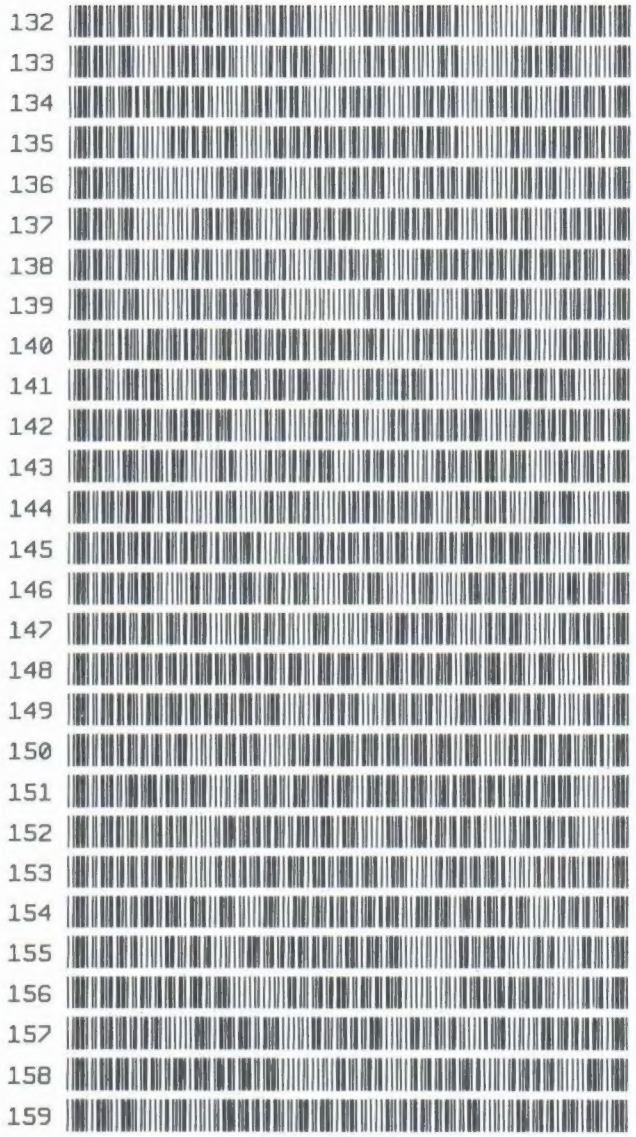
```
1000 ' *** Initialize ***
1010 ON ERROR GOTO 1040
1020 CLEAR 1000:MAXFILES=2
1030 GOTO 1050
1040 IF ERR=5 THEN RESUME NEXT
1050 ON ERROR GOTO 0
1060 RUNM "B30F9"
1070 OPEN "WAND:" FOR INPUT AS #1
1080 UC%=-1
1090 PC$="0123456789ABCDEFGHIJKLMNOPQRST
UVWXYZabcdefghijklmnopqrstuvwxyz- $+"
1100 DIM RW$(36)
1110 ER$(1)="You must scan line 1 first!
"
1120 ER$(2)="You've SKIPPED a line!"
1130 ER$(3)="You've ALREADY SCANNED this
 line!"
1140 ER$(4)="Code not PCM2/39 format!"
1150 ER$(5)="Command not applicable here
!"
1160 ER$(6)="You cannot skip this line!"
1170 ER$(7)="Selected resume file not in
 computer!"
1180 ' *** Read Reserved Words List ***
1190 DATA BEEP,CLEAR,CLOSE,DATA,DEFDBL,D
EFINT,DEFNG,DEFSTR,ELSE,GOSUB,GOTO
1200 DATA INKEY$,INPUT,INSTR(,LCOPY,LEFT
$(,LINE(,LOADM,LPRINT,USING,MAXFILES
1210 DATA MID$(,NEXT,PEEK,POKE,POWER,PRE
SET(,PRINT,READ,RESTORE,RETURN,RIGHT$(
1220 DATA SOUND,SPACE$(,STRING$(,THEN
1230 FOR I%=1 TO 36:READ RW$(I%):NEXT I%
1240 ' *** Procedure Begins Here ***
1250 CLS:PRINT@44,"PCM Bar Code Program
Reader v2.1"
1260 LINE(20,4)-(219,18),1,B:LINE(22,6)-
(217,16),1,B
1270 NN%=1
1280 GOSUB 1660:IF ER%>0 THEN GOSUB 1620
:GOTO 1280
1290 IF LL%=0 AND INSTR("YN",IL$)>0 THEN
 ER%=5:GOSUB 1620:GOTO 1280
1300 IF LL%=0 THEN ON INSTR("ALSR",IL$)
GOTO 1820,1890,1980,2050
1310 IF LL%=1295 THEN 1350
1320 IF LL%<>NN% AND NN%=1 THEN ER%=1:GO
```

```
SUB 1620:GOTO 1280
1330 IF LL%<NN% THEN ER%=3:GOSUB 1620:GO
TO 1280
1340 IF LL%>NN% AND NN%>1 THEN ER%=2:GOS
UB 1620:GOTO 1280
1350 IL$=RIGHT$(IL$,19)
1360 IF LL%=1 AND NN%>0 THEN GOSUB 1780
1370 CL$=CL$+IL$
1380 FOR I%=1 TO LEN(CL$)
1390     CH$=MID$(CL$,I%,1)
1400     IF CH$="%" THEN GOSUB 1510:IF NL
% THEN 1470 ELSE GOTO 1440
1410     IF CH$="/" THEN GOSUB 1550:IF NL
% THEN 1470 ELSE GOTO 1440
1420     IF CH$="." THEN UC%=NOT(UC%):GOT
O 1450
1430     IF CH$=>"A" AND CH$<="Z" AND NOT
(UC%) THEN CH$=CHR$(ASC(CH$)+32)
1440     XX$=XX$+CH$:IF RIGHT$(XX$,1)=CHR
$(13) THEN PRINT#2,XX$;:XX$="":UC%=-1
1450 NEXT I%
1460 CL$=""
1470 PRINT@200,SPACE$(80);
1480 IF LL%<>1295 THEN NN%=LL%+1:GOTO 12
80
1490 ' *** Done ***
1500 CLOSE:CALL 61807!:CLEAR 500,HIMEM:M
ENU
1510 ' *** Decode Reserved Word ***
1520 NL%=0:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="%":GOTO 1540
1530 I%=I%+1:CH$=RW$(INSTR(PC$,MID$(CL$,
I%,1)))
1540 RETURN
1550 ' *** Decode Hex and Control Charac
ters ***
1560 NL%=0:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="/":GOTO 1610
1570 I%=I%+1:IF INSTR("/%.",MID$(CL$,I%,
1))>0 THEN CH$=MID$(CL$,I%,1):GOTO 1610
1580 IF I%>LEN(CL$)-1 THEN NL%=-1:CL$=RI
GHT$(CL$,2):GOTO 1610
1590 HX$=MID$(CL$,I%,2):CH$=CHR$((INSTR(
"0123456789ABCDEF",LEFT$(HX$,1))-1)*16+I
NSTR("0123456789ABCDEF",RIGHT$(HX$,1))-1
)
1600 I%=I%+1
1610 RETURN
1620 ' *** Error Codes ***
1630 SOUND 5000,10:SOUND 8000,10:SOUND 5
000,10
1640 PRINT@220-.5*LEN(ER$(ER%)),ER$(ER%)
;
1650 RETURN
1660 ' *** Get Code Line ***
1670 PRINT@173,"";:PRINT USING "Scan lin
e ###";NN%
1680 IF NN%=-1 THEN PRINT@173,"Scan any
line":GOTO 1700
1690 SOUND 500,5
1700 INPUT#1,IL$:ER%=0
1710 FOR I%=1 TO LEN(IL$)
1720 IF MID$(IL$,I%,1)="!" THEN MID$(IL$
,I%,1)="."
1730 NEXT I%
1740 IF LEN(IL$)<>1 AND LEN(IL$)<>21 THE
N ER%=4:RETURN
1750 IF LEN(IL$)=1 THEN LL%=0:RETURN
1760 LL$=LEFT$(IL$,2):LL%=(INSTR("012345
6789ABCDEFGHIJKLMNOPQRSTUVWXYZ",LEFT$(LL
$,1))-1)*36+INSTR("0123456789ABCDEFGHIJK
LMNOPQRSTUVWXYZ",RIGHT$(LL$,1))-1
1770 RETURN
1780 ' *** Open Program File ***
1790 PN$=LEFT$(IL$,6):IL$=RIGHT$(IL$,LEN
(IL$)-6)
1800 OPEN PN$ FOR OUTPUT AS #2
1810 RETURN
1820   *** Abort ***
1830 BEEP:BEEP:BEEP
1840 PRINT@209,"ABORT!  Are you sure?";
1850 INPUT#1,AN$
1860 IF INSTR("YN",AN$)=0 THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 1850
1870 PRINT@200,SPACE$(80);
1880 IF AN$="Y" THEN CLOSE:KILL PN$+".DO
":GOTO 1490 ELSE GOTO 1280
1890 ' *** Skip Line ***
1900 IF NN%=1 THEN ER%=6:GOSUB 1620:GOTO
 1280
1910 BEEP:BEEP:BEEP
1920 PRINT@210,"SKIP!  Are you sure?"
1930 INPUT#1,AN$
1940 IF INSTR("YN",AN$)=0 THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 1930
1950 PRINT@200,SPACE$(80);
1960 IF AN$="Y" THEN NN%=NN%+1
1970 GOTO 1280
1980 ' *** Stop & Save ***
1990 BEEP:BEEP:BEEP
2000 PRINT@207,"STOP & SAVE! Are you sur
e?";
2010 INPUT#1,AN$
2020 IF INSTR("YN",AN$)=0 THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 2010
2030 PRINT@200,SPACE$(80);
2040 IF AN$="Y" THEN 1490 ELSE GOTO 1280
2050 ' *** Resume ***
2060 IF NN%<>1 THEN ER%=5:GOSUB 1620:GOT
O 1280
2070 PRINT@254,"Resume Mode";
2080 NN%=1:GOSUB 1660
2090 IF LL%=0 THEN ER%=5 ELSE IF LL%<>1
THEN ER%=1
2100 IF ER%>0 THEN GOSUB 1620:GOTO 2060
2110 PN$=MID$(IL$,3,6)
2120 ON ERROR GOTO 2140
2130 OPEN PN$ FOR INPUT AS #2:GOTO 2170
```

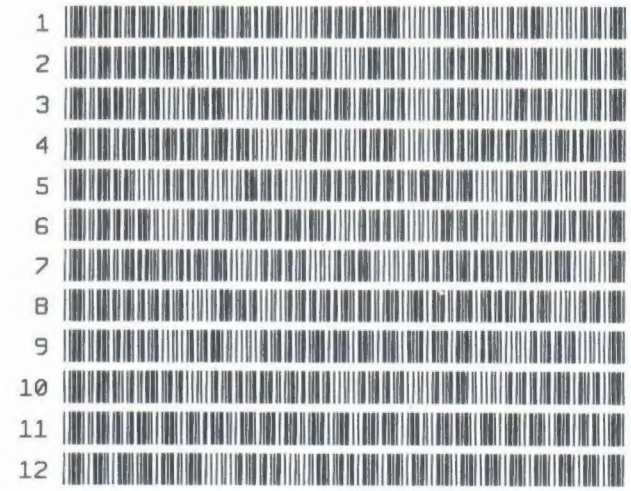
```
2140 RESUME 2150
2150 CLOSE #2
2160 ER%=7:GOSUB 1620:GOTO 1270
2170 CLOSE #2:OPEN PN$ FOR APPEND AS #2
2180 NN%=-1:GOTO 1280
```

TESTM (FROM PAGE 68)





SAMPL.DO (FROM PAGE 68)



ALABAMA	
Birmingham	Jefferson News Co.
Florence	Anderson News Co.
Madison	Madison Books
ALASKA	
Fairbanks	Fairbanks News Agency
ARKANSAS	
Fayetville	Vaughn Electronics/Radio Shack
ARIZONA	
Phoenix	Computer Pro
	Tri-Tek
Scottsdale	Softwareland Corp.
Tucson	Anderson News Co.
CALIFORNIA	
Citrus Heights	Software Plus
Half Moon Bay	Strawflower Electronics
Livermore	Software Galeria
Lompoc	L & H Electronics Emporium
North Hollywood	Levity Distributors
San Francisco	News on 24
Santa Rosa	Sawyer's News
Sunnyvale	Computer Literacy
DELAWARE	
Middletown	Delmar Co.
Wilmington	Normar Inc.—The Smoke Shop
FLORIDA	
Altamonte Springs	International Music & Recordings
Boca Raton	Software, Software, Inc.
Ft. Lauderdale	Software Connection
N. Miami Beach	Almar Bookstore
Panama City	Computer Systems Group
Pensacola	Anderson News Co.
Tallahassee	Anderson News Co.
GEORGIA	
Atlanta	Guild News Agency
Columbus	Muscogee News Co.
ILLINOIS	
Chicago	Chas. Levy Distr.
	Kroch's & Brentano's Jackson Street
	Kroch's & Brentano's Wabash Ave.
	Prairie News Agency
Oakbrook	Kroch's & Brentano's
INDIANA	
Elkhart	Elkhart City News & Book Store
Mishawaka	Carrico's Radio Shack
Scottsburg	Radio Shack of Scottsburg
IOWA	
Ames	Myers TV & Stereo Repair
Davenport	Interstate Book Store
KANSAS	
Wichita	Lloyd's Radio
KENTUCKY	
Louisville	Hawley-Cooke Bookstores
Prospect	Falsoft, Inc.
LOUISIANA	
Gretna	Computer Supply Store
Slidell	Radio Shack
MARYLAND	
Laurel	Pete Brewin
Lexington Park	Books, Etc.
MASSACHUSETTS	
Littleton	Computer Plus
Marlboro	Radio Shack
MICHIGAN	
Rochester	Rochester Book Center
Sterling Heights	Software City Programs, Inc.
Wyoming	Gerry's Book Co.
MISSOURI	
Independence	Digital Enterprises, Inc.
NEBRASKA	
Lincoln	Hobby Town
NEW HAMPSHIRE	
Manchester	Brookwrights
Petersborough	BRW Electronics-Radio Shack
West Lebanon	Verham News Corp.
NEW JERSEY	
Clinton	Micro World II
Hackensack	Total Circulation Service
Marmora	Outpost Radio Shack
Pennsville	Dave's Electronic Radio Shack
NEW MEXICO	
Alamogardo	New Horizons Computer Systems
Albuquerque	News and Bookstore
	Page One Newsstand
NEW YORK	
Hudson Falls	G.A. West & Co.
NORTH CAROLINA	
Cary	It's Just For You, Inc.
	News Center in Cary Village
Sylva	Total Data Systems
NORTH DAKOTA	
Fargo	Computer Associates
OHIO	
Cincinnati	Cinsoft
Toledo	Leo's
OKLAHOMA	
Oklahoma City	Merit Computers
PENNSYLVANIA	
Pleasant Hills	Pittsburgh Computer Store
RHODE ISLAND	
Newport	Kelly's Variety
SOUTH CAROLINA	
North Charleston	The Green Dragon
TENNESSEE	
Knoxville	Anderson News Co.
Memphis	Computer Center
Nashville	Mosko's Book Store
Smyrna	Delker Electronics, Inc.
TEXAS	
Dallas	Micro Concepts, Inc.
Ft. Worth	A & A International
VIRGINIA	
Alexandria	Alonso Book & Period
Norfolk	I-O Computers
WASHINGTON	
Kennewick	Satellite TV Specialists
WISCONSIN	
Ladysmith	Electronics, Etc.
CANADA:	
ALBERTA	
Calgary	Rainbow Software Services
Edmonton	CMD Micro
	Kelly Software Distributors
BRITISH COLUMBIA	
Sidney	Sidney Electronics
QUEBEC	
LaSalle	Messageries De Presse Benjamin
NOVA SCOTIA	
Halifax	Atlantic News

Also available at selected B.Dalton Bookseller stores

Advertisers Index

Alps 81
Axonix 15
BKI 42
Blackhawk Data Systems 20
BuddySystems 24
CDA Computer Sales 29
Computer Plus 3
Computer Solutions 84
Cryptronics Inc. 48
Custom Software 78
Delker Electronics 55
DFW COMPUTER CENTER 33
Envision Designs 29
FORT WORTH COMPUTERS 69, 83
Hard Drive Specialist 76
James River Group 40
Joe Lynn Computer Services 81
Mark Data 61
Marymac Industries 64
Micro Computer Business Service 53
Nocona Electronics 62, 65
PBJ 75
PCM Binders 72
PCMfest . . . . . IBC
Portable Computer Support Group . . . . . 16, 17, 30, 56, 70, BC
Dr. Preble's Programs 69
Purple Computing 9
Radio Shack 26, 27, 58, 59
RTX Technologies 7
Sigea Systems . . . . . IFC
Small Computer Co. 5
Software Support 13
Southwestern Digital 64

PCM Advertising and Marketing Representatives

Cindy Shackleford
Director, West Coast Office
12110 Meridian South — Suite 8
P.O. Box 73-578
Puyallup, WA 98373-0578
(206) 848-7766

Kate Tucci
Advertising Representative
9529 U.S. Highway 42
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

Join us for our very first show . . .

PCMfest®

Princeton, N.J. Oct. 11-13

You're invited to PCMfest, a great new show that's just for your Tandy computer. Sponsored by PCM, *The Personal Computing Magazine for Tandy Computer Users*, it's a wonderful way to meet and exchange information with those who share your interest in the new generation of Tandy computers.

You'll also discover the greatest variety of products ever for your computer because all of the favorites and a lot of new ones will be featured in the exhibits of PCM advertisers. Try out that new program and take it home that very day!

Along with other PCM readers, you'll meet the top national experts on your computer, including those who write for or who are written about in PCM. They will answer your questions on the spot.

PCMfest also will include a comprehensive lineup of free seminars on topics of immediate concern — and all of them designed to help you get the most out of your Tandy computer.

The Hyatt Regency - Princeton will be offering special rates ($65, single or double room) for PCMfest. The show opens Friday evening with a 7 p.m. to 10 p.m. session. It's a daytime-only show Saturday — the exhibits open at 10 a.m. and run continuously until 6 p.m. On Sunday, the exhibit hall opens at 11 a.m. and closes at 4 p.m.

Tickets may be obtained directly from PCM. We'll also send you a special reservation form so you can take advantage of the special room rate. Come to PCMfest and let's celebrate the new generation of Tandy computers!

Keynote Speaker

Bill Barden, respected author of 30 books on various computer subjects, will speak at our Saturday Breakfast at 8 a.m. (Separate tickets required.)

Free Seminars

Sam Redmon — *The Future of Portable Computing*
As co-founder of the Dallas-based Portable Computer Support Group, Sam markets a wide variety of software and hardware for the Tandy portables.

Harry Brawley — *The Portable Computer Environment*
President of Sigea Systems, he is creator of the Telecommuter integrated communications and word processing software.

Carl Oppedahl — *Computer Law and Portable Computers*
An attorney practicing law in Manhattan, Carl specializes in technological litigation. He is the author of a newly-published book titled *Advanced Programming for the Model 100.*

Bill Barden — *Computer Languages*
Discussion will include choosing the right language for your application and an overview of the most popular microcomputer languages including BASIC, C, Pascal and Assembly.

Bob Covington — *Data Communications*
A writer and editor from St. Louis, Bob is author of "The Portable Machine" series for the Tandy 100 and the "Subroutine City" series for Tandy's MS-DOS computers.

Howard Wolowitz — *Database Management*
President of Small Computer Company, Howard is one of the developers of Small's popular *Profile* and *filePro* database management systems.

Danny Humphress — *ViaNet—Tandy's New Intra-Office Communications System*
PCM managing editor and author of the "MS-DOSsier" and "dBASE Tutor" series, Danny also owns a software firm in Louisville.

RAINBOWfest Your admission to PCMfest also entitles you to attend RAINBOWfest, the highly popular show for the Tandy Color Computer which will run concurrently with PCMfest at the same location.

YES, I'm coming to Princeton! I want to save by buying tickets now at the special advance sale price.

Please send me:

_____ three-day tickets at $9 each total _____

_____ one-day tickets at $7 each total _____
(Circle one) Friday Saturday Sunday

_____ Saturday Breakfast at $12 each total _____

Handling Charge $1 _____

TOTAL ENCLOSED _____

U.S. FUNDS ONLY, PLEASE

☐ Also send me a hotel reservation card for the Hyatt Regency-Princeton ($65, single or double room).

Name (please print) _____

Address _____

City _____ State _____

Telephone _____ ZIP _____

Company _____

Charge to:
☐ VISA ☐ MasterCard ☐ American Express

Account Number _____

Exp. Date _____

Signature _____

Make checks payable to: PCM. Mail to: PCMfest, the Falsoft Building, P.O. Box 385, Prospect, KY 40059. To make reservations by phone, call: (502) 228-4492.

Advance ticket deadline: October 4, 1985. Orders received less than two weeks prior to show opening will be held for you at the door. Tickets will also be available at the door at a slightly higher price.

6 ROM BANK

plus powerful Nicad battery pack that gives 30 hours of power to your Model 100

$269

A joint PCSG/CRYPTRONICS project.

Now you can access LUCID, DISK+, WRITE ROM, RAM+ and others instantly.

The 6 ROM BANK plus battery pack lets you have the ultimate Model 100 system. When you get your unit just press in up to 6 ROMS into the sockets then push in a simple plug into your Model 100's ROM socket. The ROM bank has two posts on either end that insert into the two little holes on the underside of your Model 100.

The ROM bank props up the Model 100 at the same angle and height as those little legs you've seen. The ROM BANK itself is only about 1½" deep and it runs the width of your Model 100. It only weighs one pound. It not only installs instantly, but it pops free in a second if you need everything to lie flat in a briefcase.

Change from ROM to ROM with the touch of a thumb switch.

You can go from LUCID to WRITE to DISK+ to any other ROMS just by turning the thumb switch at the side of the ROM bank. The 6 ROM BANK is a sturdy well built construction that looks like it is a part of your Model 100.

What is also fantastic is that the ROM bank has a powerful NICAD battery and recharger built right in. This power source supplies up to 30 hours of life to your Model 100 with just 6 hours of recharge. What's nice is that it recharges right from your Model 100's power adapter. This is a quick charge system and if you need power in a hurry, you can get 6 hours of life for your Model 100 by just charging for an hour and half.

Suddenly, the Model 100 is a very powerful computer.

If you have the HOLMES portable disk drive with its powerful bundled database software package or the CRYPTRONICS 128K RAM expansion along with LUCID, WRITE ROM and DISK+ in the 6 ROM bank with its rapid rechargable NICAD power source, you have the ultimate portable system. It's all available only from PCSG.

As usual we don't want you to take our word for it. The 6 ROM BANK is sold on a 30 day trial. If you are not satisfied simply return within thirty days for a full refund. MC, VISA, AM. EX. or C.O.D.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

VISA / MASTER CARD / AM. EXP. / C.O.D.